

REFERENCE MANUAL

Version 5.0

SMS 5.0

Copyright © 1995 Brigham Young University - Engineering Computer Graphics Laboratory March 31, 1997

All Rights Reserved

Unauthorized duplication of the SMS software or user's manual is strictly prohibited.

THE BRIGHAM YOUNG UNIVERSITY ENGINEERING COMPUTER GRAPHICS LABORATORY MAKES NO WARRANTIES EITHER EXPRESS OR IMPLIED REGARDING THE PROGRAM *SMS* AND ITS FITNESS FOR ANY PARTICULAR PURPOSE OR THE VALIDITY OF THE INFORMATION CONTAINED IN THIS USER'S MANUAL

The software *SMS* is a product of the Engineering Computer Graphics Laboratory of Brigham Young University. For more information about this software and related products, call (801) 378-2812. FAX (801) 378-2478. Or write to:

Brigham Young University
Engineering Computer Graphics Laboratory
368B CB, Provo, Utah 84602

# **TABLE OF CONTENTS**

1 INTRODUCTION	1-1
1.1 Overview	1-1
1.2 COMMAND LINE ARGUMENTS.	
1.3 Modules	
1.3.1 2D Mesh Module	
1.3.2 2D Boundary Fitted Grid Module	
1.3.3 2D Scatter Point Module	
1.3.4 Map Module	1-5
1.3.5 River Module	
1.4 Data Sets	1-5
1.5 ABOUT THIS MANUAL	1-6
2 GENERAL TOOLS	2-1
2.1 SMS SCREEN	2-1
2.2 THE GRAPHICS WINDOW	2-2
2.3 TOOL PALETTE	2-2
2.3.1 The Module Palette	2-3
2.3.2 The Static Tool Palette	2-3
2.3.3 The Dynamic Tool Palette	2-4
2.3.4 The Macros	2-5
2.4 PLOT WINDOW	2-5
2.5 Edit Window	2-6
2.5.1 The Edit Box	2-6
2.5.2 The Help Window	
2.6 Menu Bar	
2.7 FILE MENU	
2.7.1 File Types	
2.7.2 New	
2.7.3 Open	
2.7.4 Save	
2.7.5 Import File	
2.7.6 Export File	
2.7.7 Get Info	
2.7.8 Save Environment	
2.7.9 Print	
2.7.10 Demo Mode	
2.7.11 Register	
2.7.12 Quit	
2.8 Edit Menu	
2.8.1 Delete	
2.8.2 Select Commands	
2.8.3 Confirm Deletions	
2.8.4 Materials	
2.9 DISPLAY MENU	2-17

2.9.1 Refresh	2-17
2.9.2 Automatic Refresh	
2.9.3 Display Options	
2.9.4 Frame Image	2-18
2.9.5 Set Window Bounds	2-18
2.9.6 Grid Options	2-18
2.9.7 Background Color	
2.9.8 Plot Window	
2.9.9 River Window	2-19
3 DATA VISUALIZATION	3-1
3.1 Browser	3-2
3.1.1 File I/O	
3.1.2 Active Data Set	
3.1.3 Elevations & Automatic Data Sets	
3.1.4 Deleting Data Sets	3-4
3.1.5 Data Set Info	3-4
3.2 Data Calculator	3-5
3.3 VECTOR OPTIONS	3-6
3.4 CONTOUR OPTIONS	3-7
3.5 COLOR RAMP OPTIONS	3-9
3.6 CONTOUR LABELS	3-10
3.7 FILM LOOPS	3-11
3.7.1 Film Loop Setup	3-11
3.7.2 Film Loop Playback	3-14
3.7.3 Saving Film Loops	3-14
3.8 GAGES	3-15
3.8.1 The Gages Dialog	
3.8.2 The Gage Tools	
3.9 THE PLOT MANAGER	3-18
4 MESH MODULE	4-1
4.1 TOOL PALETTE	4-1
4.1.1 Create Mesh Nodes	4-1
4.1.2 Select Mesh Nodes	4-2
4.1.3 Create Nodestrings	4-2
4.1.4 Select Nodestrings	4-3
4.1.5 Create Gages	4-3
4.1.6 Select Gages	4-3
4.1.7 Create Elements	4-3
4.1.8 Select Elements	4-5
4.1.9 Swap Edges	4-5
4.1.10 Merge/Split	4-5
4.1.11 Label Contours	4-5
4.2 MESH CONVERSION	4-5
4.2.1 Mesh -> Scatter Point	4-6
4.2.2 Material -> Feature	4-6
4.3 MESH DISPLAY OPTIONS	4-7
4.4 MESH GENERATION	4-8
4.5 Node Operations	4-9
4.5.1 Creating Nodes	4-9
4.5.2 Find Operations	4-12
153 Editing Nodes	1 13

4.5.4 Deleting Nodes	4-13
4.5.5 Interpolating Nodal Boundary Conditions	
4.6 Transforming Mesh	
4.7 Element Operations	
4.7.1 Triangulation	
4.7.2 Element Options	
4.7.3 Rectangular Patches	
4.7.4 Triangular Patches	
4.7.5 Find Operations	
4.7.6 Breaklines	
4.7.7 Merge Triangles	
4.7.8 Split Quadrilaterals	
4.7.9 Element Conversion	
4.7.10 Refining Elements.	
4.7.11 Relaxing Elements	
4.7.12 Smooth Edges	
4.7.13 Renumbering	
4.8 Assign Materials	
4.9 Model Menus	
5 SCATTER POINT MODULE	5-1
5.1 SCATTER POINT SETS	5-1
5.2 Inputting Sets	
5.3 SAVING SETS	
5.4 Tool Palette	
5.4.1 Select Scatter Point	
5.4.2 Select Scatter Point Set	
5.5 DISPLAY OPTIONS	
5.6 SCATTER POINT CONVERSION.	
5.6.1 Scatter Points -> Mesh Nodes	
5.7 Interpolation	
5.7.1 Interpolation From Scatter Point Sets	
5.7.2 Interpolation Options	5-4
5.7.3 Linear Interpolation	
5.7.4 Inverse Distance Weighted Interpolation	
5.7.5 Clough-Tocher Interpolation	
5.7.6 Natural Neighbor Interpolation	
6 MAP MODULE	6-1
6.1 Feature Objects	6-2
6.1.1 Feature Object Types	
6.1.2 Feature Object Tools	
6.1.3 Build Polygon	
6.1.4 Clean	
6.1.5 Vertex <-> Node	
6.1.6 Redistribute Vertices	
6.1.7 Coverages	
6.1.8 Display Options	
6.1.9 Assigning and Editing Attributes	
6.2 2D MESH COVERAGE	
6.2.1 Point/Node Attributes	
6.2.2 Arc Attributes Dialog	
6.2.3 Polygon Attributes Dialog	
VI. (1)   1. VILYYOU   [ALLIUMEN DIMIUY	

6.2.4 Constructing 2D Meshes	6-20
6.2.5 2D Mesh Attributes	
6.2.6 Map -> 2D Mesh	6-22
6.3 DRAWING OBJECTS	6-23
6.3.1 Drawing Object Tools	6-24
6.3.2 Display Attributes	6-25
6.3.3 Display Options	6-27
6.3.4 Drawing Order	6-27
6.4 IMAGES	6-28
6.4.1 Importing an Image	6-28
6.4.2 Registering an Image	6-28
6.4.3 Resampling an Image	6-30
6.4.4 Fit Entire Image	
6.4.5 Deleting Images	6-31
6.4.6 Exporting the Resampled Region	
6.4.7 Export TIFF vs. Save Image	
6.5 DXF FILES	
6.5.1 Importing DXF Files	
6.5.2 Display Options	
6.5.3 DXF -> Feature Objects	
6.5.4 DXF -> Scatter Points	
6.5.5 Deleting DXF Files	
6.6 READING AND SAVING MAP FILES	6-35
7 RIVER MODULE	7-1
7.1 Create Station	7-2
7.2 SELECT STATION.	
7.3 CREATE SECTION TOOLS.	
7.3.1 Create Bridge Section.	
7.3.2 Create Road Section	
7.3.3 Create Culvert Section	
7.4 SELECT SECTION	
8 RMA2 INTERFACE	
8.1 OPEN GEOMETRY	
8.3 OPEN BC	
8.4 SAVE BC	
8.5.1 Head Boundary Conditions	
8.5.2 Velocity Boundary Conditions	
8.6 ASSIGN STRING BC	
8.6.1 Flow Boundary Conditions.	
8.6.2 Head Boundary Conditions	
8.7 DELETE BC	
8.8 ADD GC String	
8.9 RMA2 MATERIALS	
8.10 MODEL CHECK	
8.11 GLOBAL BC CONTROL	
8.11.1 Job Title	
8.11.2 File Control	
8.11.3 Iteration Control	
8 11 4 Computation Time Control	

8.11.5 Units Control	8-11
8.11.6 Other Options Control	8-11
8.12 OPTIONAL BC CONTROL	8-11
8.12.1 Machine Type	8-12
8.12.2 Geometry Modifications	8-12
8.12.3 Wetting and Drying	8-13
8.12.4 Peclet Number Control	
8.12.5 Echo Control	
8.13 DISPLAY OPTIONS	8-13
9 SED2D-WES INTERFACE	9-1
9.1 SED2D-WES FILE I/O	9-1
9.1.1 New Simulation	9-2
9.1.2 Open Simulation	9-2
9.1.3 Save Simulation	9-2
9.2 GLOBAL PARAMETERS	9-2
9.2.1 Bed Type	9-3
9.2.2 Diffusion Coefficients	
9.2.3 Initial concentration	
9.2.4 Settling velocity	
9.3 LOCAL PARAMETERS	
9.4 BC CONCENTRATIONS	
9.5 MODEL CONTROL	
9.6 PRINT CONTROL	
9.7 SED2D-WES DISPLAY OPTIONS	
9.8 Model Checker	
10 HIVEL INTERFACE	
10.1 New Simulation	
10.2 OPEN SIMULATION	
10.3 SAVE SIMULATION	
10.4 BUILD HOT START	
10.5 Assign BC	
10.5.1 Inflow Boundary	
10.5.2 Outflow Boundary	
10.6 DELETE BC	
10.7 MODEL CONTROL	
10.7.1 Job Titles	10-5
10.7.2 Turbulence Coefficients	10-5

11.2 SAVE SIMULATION	11-3
11.3 NODAL BOUNDARY CONDITIONS	11-4
11.3.1 Specify x/Tangent Condition	11-5
11.3.2 Specify y/Normal Condition	11-6
11.3.3 Specify Water Surface or Source/Sink	11-6
11.4 BOUNDARY SECTION	
11.4.1 Flow	11-6
11.4.2 Water Surface Elevation	11-7
11.4.3 Rating Curve/Friction Slope	
11.5 INITIAL CONDITIONS	
11.6 WIND CONDITIONS	
11.7 Weirs	11-9
11.8 CULVERTS.	
11.9 Drop Inlets	
11.10 Piers	11-10
11.11 Node Ceilings.	11-11
11.12 Flux String	
11.13 MATERIAL PROPERTIES	
11.14 MODEL CHECK	
11.15 FESWMS CONTROL	
11.15.1 Project Title	
11.15.2 Save File Options	
11.15.3 Output Format	
11.15.4 Solution Type	
11.15.5 Units	
11.15.6 Bottom Stresses.	
11.15.7 Slip Conditions	
11.15.8 Higher Order Integration	
11.15.9 Control Buttons	
11.16 FESWMS DISPLAY OPTIONS	
12 WSPRO INTERFACE	12-1
12.1 New Simulation	12-2
12.2 OPEN SIMULATION	12-2
12.3 SAVE SIMULATION	12-2
12.4 EDIT SECTION	12-3
12.4.1 Plot Tools	12-3
12.4.2 View Tools	
12.4.3 Section Translation Tools	
12.4.4 Section Edit Fields	
12.4.5 Section Editor Buttons	
12.5 VIEW DATA FILE.	12-10
12.6 ROUGHNESS PARAMETERS	12-11
12.7 WSPRO Run Control	12-12
12.8 JOB PARAMETERS.	12-13
12.9 DISPLAY OPTIONS	
12.10 Model Check.	
12.11 Run WSPRO.	
13 XY SERIES EDITOR	
13.1 XY Series List	12 1
13.2 The XY Edit Fields	
13.2.1 Delate	

13.2.2 Interpolate	13-3
13.2.3 Update	
13.2.4 Insert	
13.2.5 Compress	
13.2.6 XY Options	
13.3 THE XY SERIES PLOT	13-4
13.3.1 The Plot Tools	
13.3.2 The Plot Macros	13-5
14 FILE FORMATS	14-1
14.1 SMS SUPER FILES	14-2
14.2 2D Mesh Files	14-3
14.3 2D SCATTER POINT FILES	14-5
14.4 ASCII DATA SET FILES	14-8
14.5 Binary Data Set Files	14-12
14.6 ASCII Scalar Data Set Files (version 4)	14-17
14.7 ASCII VECTOR DATA SET FILES (VERSION 4)	14-19
14.8 Binary Scalar Data Set Files (version 4)	14-21
14.9 BINARY VECTOR DATA SET FILES (VERSION 4)	
14.10 Gage Files	
14.11 Map Files	14-29
14.11.1 General	14-30
14.11.2 Feature Objects	
14.11.3 Feature Object Attributes	
14.11.4 Drawing Objects	
14.12 Image Files	
14.13 MESH FROM POLYGON FILES	
14.14 XY Series Files	
14.15 TIN Files	
14.16 Material Files	
14.17 XYZ Files	14-52
15 REFERENCES	15-1
16 INDEX	16-1

CHAPTER 1

## Introduction

*SMS* is a comprehensive environment for hydrodynamic modeling. It was developed by the Engineering Computer Graphics Laboratory at Brigham Young University in cooperation with the U.S. Army Corps of Engineers Waterways Experiment Station (WES), and the U.S. Federal Highway Administration (FHWA).

#### 1.1 Overview

SMS is a pre- and post-processor for surface water modeling and analysis. It includes two-dimensional finite element, two-dimensional finite difference and one-dimensional backwater modeling tools. Interfaces have been specifically designed to be used in conjunction with the TABS-MD suite of programs maintained by WES. Including: GFGEN, RMA2, RMA4, SED2D-WES and HIVEL2D. Comprehensive interfaces have also been developed for facilitating the use of the FHWA commissioned analysis package FESWMS and WSPRO.

These hydrodynamic modeling programs will calculate water surface elevations and flow velocities for shallow water flow problems and support both a steady-state and dynamic model. Additional tools are provided in *SMS* to support the modeling of contaminant migration and sediment transport.

The finite element meshes or cross sectional properties along with associated boundary conditions necessary for analysis may be created within *SMS* and saved to model specific files. These files are used to perform the hydrodynamic, contaminant migration, and sediment transport analyses. The resulting solution files, which contain the water surface elevation, flow velocity, contaminant concentration, sediment concentration or other functional data at each node of the mesh, can be read

into *SMS* to generate profiles and cross sectional plots, two -dimensional vector plots, color-shaded contour plots, time variant curve plots, and dynamic animation sequences.

*SMS* can also be used as a pre- and post-processor for other finite element or finite difference programs provided the programs can be made to read and write files in a supported file format. *SMS* is well-suited for the construction of large, complex meshes (several thousand elements) of arbitrary shape. Sample meshes are shown in Figure 1.1 and Figure 1.2.

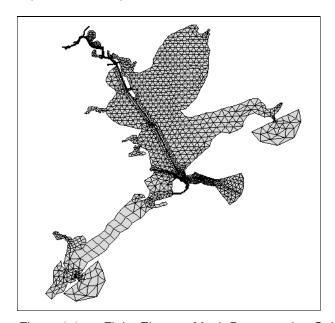


Figure 1.1 Finite Element Mesh Representing Galveston Bay.

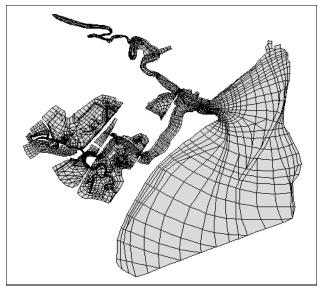


Figure 1.2 Finite Element Mesh Representing Saugus River Estuary.

This document is meant to be a reference manual for *SMS* only. Details on the analysis software is contained in separate documents. It is assumed that the user is familiar with the overall modeling process and terms defined in the appropriate supporting reference manual.

*SMS* was designed as a comprehensive hydrodynamic modeling system. As part of an ongoing effort, other analysis models will be supported in future versions.

## 1.2 Command Line Arguments

SMS provides for a few command line arguments, which facilitate the use of SMS. These arguments are simply entered after the command for SMS at the command line. Some of the arguments are not available on the PC and are so noted.

The following commands are available:

#### help

The help command displays usage information SMS commands.

Usage: sms -help

about (workstation only)

The about command displays SMS copyright and vendor information.

Usage: sms -about

dm <module>

The default module command is used to specify the default module.

Possible values include: mesh, scat, map, river.

Usage: sms -dm mesh

**r** <file spec> (workstation only)

The resource directory command is used to specify the path name for the resource directory.

Usage: sms -r /ecgl/geos/sms/bin

ini <file spec> (workstation only)

The initialization path command is used to specify the path where the initialization file should be read and saved.

Usage: sms -ini /ecgl/User

**tmp** <file spec> (workstation only)

The temp path command is used to specify the path where temporary files should be saved.

Usage: sms -tmp /ecgl/User

### 1.3 Modules

The interface for *SMS* is divided into five separate modules. A module is provided for each of the basic procedures supported by *SMS*. As the user switches from one module to another, the *Tool Palette* and the menus change. This allows you to focus only on the tools and commands related to the process you are currently working on. Switching from one module to another can be done instantaneously to facilitate the simultaneous use of several processes when necessary. Multiple analysis models may be applicable to each module. Commands specific to an analysis model reside in a menu corresponding to that model. The following modules are supported in *SMS*.

#### 1.3.1 2D Mesh Module

The *Mesh Module* is used to manipulate 2D finite element meshes. A variety of tools are provided for mesh generation and mesh editing. In *SMS*, 2D meshes, or element networks are used as the basis for analysis for both the TABS suite of software and the *FESWMS* analysis package. After an analysis, output data at each node of the mesh can be used to generate contour and fringe plots to represent the solution. Multiple time steps from time variant solutions can be strung together to form an animation of the dynamic solution.

## 1.3.2 2D Boundary Fitted Grid Module

The *Grid Module* will be used to construct 2D boundary fitted grids. This module will be available in a future release of *SMS*. This module has been included in this version to facilitate future expansion.

#### 1.3.3 2D Scatter Point Module

The 2D Scatter Point Module is used to interpolate from groups of 2D scattered data points to any of the other data types. For example, the user may gather field data points representing the bathymetry of the region to be modeled. The elevation data from these points can be interpolated to a well structured set of elements to create the bathymetry of the entire mesh. The 2D Scatter Point Module can be used to

interpolate from a set of scattered xy points representing the empirical data to a finite element mesh. A variety of interpolation schemes are supported.

## 1.3.4 Map Module

The *Map Module* is used to manipulate four types of objects: DXF objects, image objects, drawing objects, and feature objects. The first three objects: DXF objects, image objects, and drawing objects are primarily used as graphical tools to enhance the development and presentation of a model. DXF objects consist of drawings imported from standard CAD packages such as *Autocad* or *Microstation*. Drawing objects are a simple set of tools that are used to draw text, lines, polylines, arrows, rectangles, etc., to add annotation to the graphical representation of a model. Image objects are digital images representing aerial photos or scanned maps in the form of TIFF files. The fourth type of objects are patterned after the data model used by geographic information systems (GIS) such as ARC/INFO. Once a conceptual model is constructed, it can automatically be converted into a mesh model.

#### 1.3.5 River Module

The *River Module* is used to construct 1D river profiles for step backwater models. Tools are provided in this module for creating a "tree" of data to describe the river being modeled. Currently, only the *WSPRO* model is supported in the river module. Tools are provided for the creation of river sections. Properties can be assigned to sections, and structures such as bridges and culverts can be added, and step backwater analysis performed.

#### 1.4 Data Sets

An important feature of *SMS* is that the interface to each of the separate modules is designed in a consistent fashion. Once the user becomes familiar with the interface to one of the modules, the other modules can be used immediately with little additional training. In order to help provide a consistent interface, the concept of generic data sets is used in *SMS*. A data set is a set of scalar or vector values associated with an object. Each data set can be either steady-state or dynamic (multiple values representing the data values at different points in time). Meshes, grids, and scatter point sets all have an associated list of scalar data sets and a list of vector data sets. Each set has a single vector or scalar value for each node, cell, or scatter point.

A data set is a set of numerical values for each node in the mesh or point in the grid. A data set of scalar values is used to represent quantities such as the water surface elevation " computed by a hydrodynamic model or empirical values used as initial conditions for input to a dynamic model. A data set of vector values is used to represent quantities such as flow velocities. Data sets can be imported from a file,

created by interpolating from a scatter point set, or computed using other datasets, constants and mathematical operators.

For example, to compare the difference in the solutions from two separate simulations on the same finite element mesh, the two solutions can be input as data sets and the data set calculator can be used to compute the absolute value of the difference between the two data sets. The resulting data set can be contoured just like any other data set.

#### 1.5 About This Manual

This reference manual has been designed to parallel the modular concept used in *SMS*. Chapters 2 and 3 describe the portions of *SMS* that are common to all modules. These chapters should be read regardless of which module the user intends to use. Chapters 4, 5, 6 and 7 describe the *Mesh*, *Scatterpoint*, *Map and River Modules* supported by *SMS*. Chapters 8, 9, 10, 11 and 12 describe the interfaces to *RMA2*, *SED2D-WES*, HIVEL2D, *FESWMS* and WSPRO models respectively. Chapter 13 discusses the *XY Series Editor*, and Chapter 14 describes the *File Formats* used in SMS.

This manual applies to both the Unix version and the MS Windows version of *SMS*. Most of the features are identical between the two versions. Any differences in the two versions are noted explicitly in the documentation.

CHAPTER 2

## General Tools

The interface to *SMS* has been designed in a modular fashion. Five separate modules representing different data types are supported. As the user switches from one module to another, a portion of the interface (menu commands, tools, etc.) changes and a portion of the interface remains unchanged. The part that remains the same provides access to general tools that are used by all of the modules. These tools are described in this chapter.

## 2.1 SMS Screen

The SMS screen is divided into five main sections: the Graphics Window, the Plot Window, the Tool Palette, the Edit Window, and the Menu Bar (see Figure 2.1).

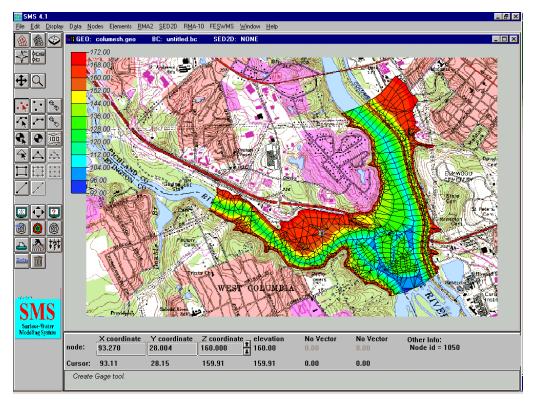


Figure 2.1 The SMS Screen.

## 2.2 The Graphics Window

The primary graphical input and output for two dimensional entities in *SMS* takes place in the *Graphics Window*. The action taken when the user interacts with the *Graphics Window* depends on which tool in the *Tool Palette* is selected.

## 2.3 Tool Palette

The *Tool Palette* is divided into four parts as shown in Figure 2.2.

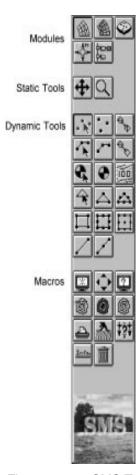


Figure 2.2 SMS Tool Palette.

#### 2.3.1 The Module Palette

The *Module Palette* is used to switch between modules. Only one module is active at any given time. However, the data associated with a module (e.g. a 2D finite element mesh) is preserved when the user switches to a different module. Activating a module simply changes the set of available tools and menu commands.

#### 2.3.2 The Static Tool Palette

The tools which are available in every module are located in the *Static Tool Palette*. These tools are tools for basic operations such as panning and zooming. The static tools are as follows:

## The Pan Tool

The *Pan* tool is used to pan the viewing area of the *Graphics Window*. When the *Pan* tool is active, clicking the mouse in the *Graphics Window* has the following results:

- If a point is clicked, the viewing area is shifted so that the point clicked corresponds to the center of the window.
- If the cursor is dragged while holding the mouse button down, the viewing
  area is shifted to simulate moving the image the direction and distance
  specified by the line defined while dragging the cursor. The image isn't
  updated until the mouse button is released.
- Holding down both the CONTROL key and the mouse button while dragging the cursor in the Graphics Window causes the moving image to be updated dynamically.

## The Zoom Tool

The viewing area can be magnified or shrunk using the *Zoom* tool. When this tool is active, the following actions can be used to redefine the viewing area of the *Graphics Window*:

- A rectangle can be dragged around a portion of the display to zoom in on a
  particular region. The display is refreshed and the area inside the rectangle is
  expanded to fill the entire screen.
- If a point is clicked, the display is zoomed in around the point by a factor of two.
- A rectangle can be dragged around a portion of the display while the *SHIFT* key is held down to zoom out to a particular region. The display is refreshed and the entire screen area shrinks to the inside of the rectangle.
- If a point is clicked while the *SHIFT* key is held down, the display is zoomed out about that point by a factor of two.

#### 2.3.3 The Dynamic Tool Palette

When the active module is changed, the tools in the *Dynamic Tool Palette* change to the set of tools associated with the selected module. Each module has a separate set of tools.

#### **Selection Tools**

Many of the module specific tools in the dynamic portion of the *Tool Palette* are selection tools (tools used to select objects such as nodes). It is necessary to first select some objects before issuing many of the commands in *SMS*. For example, to delete a set of elements in the *Mesh Module*, the *Select Elements* tool is chosen, the set of elements to be deleted are selected, and the *Delete* command is selected from the *Edit* menu, the macro panel or the *DELETE* key is hit. Specific tools are described individually in later chapters.

Most of the selection tools follow a standard selection protocol. One item can be selected by clicking on the item. When a new item is selected, any other selected items are unselected.

In many cases, multiple items need to be selected. If the *SHIFT* key is held down while clicking on individual items, the items are added to the set of selected items. A previously selected item can be unselected by holding down the *SHIFT* key and clicking on it again. This removes the item from the set of selected items without affecting other selected items. Multiple objects can also be selected by dragging a box around the items to be selected. In some cases, other selection mechanisms are implemented for versatility. For example, if the *CONTROL* key is held down while dragging to select elements, a arrow is drawn which selects all elements which intersect the arrow.

Other commands for selecting multiple objects such as *Select With Poly* can be found in the *Edit* menu and is described later in this chapter.

#### 2.3.4 The Macros

Many of the more frequently used menu commands can be accessed through the macro buttons in the lower part of the *Tool Palette*. These buttons essentially serve as shortcuts to menu commands.

### 2.4 Plot Window

The *Plot Window* is used to plot 2D curves. These curves may represent computed quantities such as concentration or elevation vs. time, or steady state quantities such as velocity along a profile or cross sectional slice.

For time variant curves the values are interpolated from computational meshes and grids to gage or observation points defined by the user.

Profiles/cross sections plots are made along node strings or observation lines, or along river sections.

The *Plot Window* is displayed at startup only if the river module is selected as default (see sec. 1.2). The display of the *Plot Window* is toggled using the show/hide plot window option in the display menu.

### 2.5 Edit Window

There are two sections in the *Edit Window*: the *Edit Box* and the *Help Window*.

#### 2.5.1 The Edit Box

The *Edit Box* is on the top half of the *Edit Window*. The top row of the edit box is used to edit the coordinates of the selected mesh node, grid node, or scatter point. The coordinates are changed by typing in new values and hitting the *ENTER* or *TAB* key. If more than one node is selected, only the z-value is available for editing. To the right of the z coordinate edit box are two arrows for interactively adjusting the z coordinate of the selected object. Entering a new value here will modify the bathymetry of each of the selected nodes. This allows the user to quickly model a feature such as a dredged channel.

The second row of numbers in the edit box are used to display the coordinates of the cursor. The z coordinate corresponds to an interpolated elevation value from either the mesh, or the grid, depending on which module is active.

The scalar and vector data values associated with the selected object and the cursor are displayed to the right of the coordinate values.

Other information about the selected object is displayed in the right side of the edit window. This information displayed are features such as the ID of a selected node, element type of selected element, nodestring type, etc.

### 2.5.2 The Help Window

The *Help Window* on the bottom of the *Edit Window* is where *SMS* provides the user with context sensitive help messages or information regarding the current operation. As the cursor is moved over tools, macros, or menu items, a description of the item appears in the *Help Window*. In addition, help messages appear in the *Help Window* as the cursor is passed over items in dialog boxes. The Windows help utility can also be accessed through the *Help* menu.

#### 2.6 Menu Bar

The commands in *SMS* are accessed through pull down menus located in the menu bar. Each menu can be accessed with the mouse or by pressing the highlighted letter in the menu title. Once a menu is visible the individual commands can be selected with the mouse or by holding down the *ALT* key and pressing the highlighted letter in the menu command.

When the active module is changed, the menus change to a set of menus associated with the selected module. The first three menus (*File, Edit, Display*) are the same for every module. The remaining menus are dependent on the selected module.

## 2.7 File Menu

The *File* menu is one of the standard menus and is available in all of the modules. The commands in the file menu are used for opening and saving SMS generic files, for printing, and to quit the program.

## 2.7.1 File Types

The file types supported in the *File* menu correspond to the generic file types only. The commands for opening and saving files associated with specific numerical models such as *RMA2* and *FESWMS* are found in the menus associated with that numerical model.

The generic file types are as follows:

- **2D Scatter Point File:** File containing one or more sets of 2D scatter points. Scatter point files can be saved using either the XY or XYD format described in Section 14.3.
- **Material File:** File containing definition of general materials parameters. Since materials are different in the different models, this generic file consist of display parameters such as color, name and ID.
- **Image file:** The file name and registration data related to a TIFF image used for background display. The actual TIFF image is saved in a separate file. This file is created by importing and registering a TIFF image and then saving the registration file.
- **Map file:** Feature objects, drawing objects and boundary conditions placed on the feature objects are saved in this file.
- **2D Mesh file:** This is a generic geometry file. It does not include boundary conditions.
- **Data Set file:** This is a generic geometry file. It may consist of scatter point bathymetry, or other sets of data.
- **Settings file:** This file saves the current settings of the program (display options, defaults, etc.). See section 2.7.8

All of the generic files are ASCII text files. The first item in each of these files is a keyword signifying the file type. The formats for these files are described in Chapter 14.

#### 2.7.2 New

The *New* command deletes all data associated with all data types and all modules. It resets the status of the program to the default state that is set when the program is first launched. This command should be selected when an entirely new modeling problem is started.

### 2.7.3 Open

The *Open* command is used to read in one or more of the generic file types. This command brings up a file browser from which a single file is selected. The program reads the keyword at the beginning of the file to determine the type of file selected and the appropriate routine is used to read the input from the file.

#### 2.7.4 Save

The *Save* command is used to save the generic data types to disk. This command brings up *Save Item* dialog (see Figure 2.3) that contains a check box, a file browser button, and a filename for each of the basic file types.

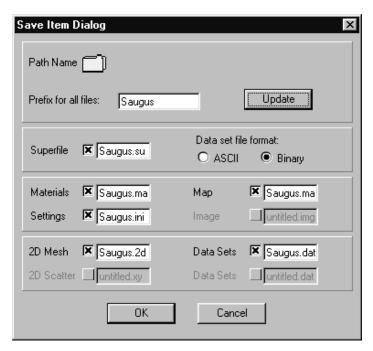


Figure 2.3 The Save Item Dialog.

If a data type does not currently exist in memory, the check box, button, and filename for that data type is dimmed. The check box can be toggled off or on to indicate whether or not the data type is to be saved to disk. The button is used to bring up the standard file browser to select the target directory and the filename. A superfile can be generated that contains a list of all of the files that are specified in the dialog. The superfile can then be used to open all of the files at once using the *Open File* command.

#### 2.7.5 Import File

The Import command will bring up the *Select Import Format* dialog that will allow the user to import data from one of several non-standard data files. The formats of these non-standard files are described in Chapter 14 or in model specific documentation. SMS asks the user whether this data should be appended to the current mesh data, or if it should replace the mesh.

A user may create a model in sections. After each section is created, they can be imported and pieced together. The *Transform Mesh* command (page 4-13) allows the user to select each piece and adjust it in relation to the other sections of the mesh. The individual sections of geometry may be contained in *any* of the import file formats, such as *FESWMS* files, TABS files and 2D Mesh files. Currently supported data file formats include:

#### **TIN File**

A TIN file stores the data for a triangulated irregular network. When *SMS* imports a TIN file, each triangle is converted to a triangular element. The type of element depends on the current element types in *SMS*. These new elements are added to the 2D finite element mesh data. This can be a useful way of generating a background mesh. The use of a background mesh is discussed in Section 4.4.

#### **XYZ File**

An XYZ file contains a list of coordinate points. When SMS imports an XYZ file the data points are converted to nodes. This provides a convenient way to import a set of points for the mesh construction operations described in Chapter 4

#### **TIFF** image File

An *TIFF* file is an image that can be imported into SMS. *TIFF* files of aerial photographs, and scanned or digitized maps can be imported as a background display for aid in digitization, visualization and model construction. (see section 6.4)

#### **DXF File**

*DXF* files are output by computer drafting programs. The format is defined by AutoDesk's AutoCAD®. These files can be imported through the file menu. They

include points, lines, and polylines. *DXF* files can serve as a background to aid in model construction or can be converted to feature objects in the *Map Module*.

#### **GFGEN** Geometry File

A *GFGEN* file contain the geometric description of a mesh along with some other mesh renumbering information. When importing this type of file, all information other than the geometry is ignored. Therefore, import is recommended only for merging sections of mesh contained in separate *GFGEN* files.

#### FESWMS File

A *FESWMS* file contains the geometric description of a mesh in addition to other parameters to control the analysis model. When importing this type of file, all information other than the geometry is ignored. Therefore, import is recommended only for merging sections of mesh contained in separated *FESWMS* files.

#### 2D Mesh File

A 2D Mesh file is a generic file format that can store the nodes and elements of a mesh or mesh section. It is generally better to store a mesh in the model specific file format such as an *GFGEN* ".geo" file. However, if the model to be used is unknown, or a section of mesh has been created by another utility and is to be imported by *SMS*, a 2D Mesh file may be used.

#### 2.7.6 Export File

*SMS* supports an *Export* command to communicate or interface with programs that utilize similar data. *SMS* supports geometric data in four formats for this purpose and image data in one format. The geometric data formats supported via the export command include:

- TIN (Triangulated Irregular Network): This is nodes connected to form Delaunay criterion triangles. This format is used by such programs as the *Watershed Modeling System (WMS)*, which uses TINs to model terrain.
- QUAD4: This format includes four sided elements for use with the QUAD4 analysis package.
- 2DMESH: This is a generic file format defined by the Engineering Computer Graphics Laboratory (see Chapter 14). The *Groundwater Modeling System* (*GMS*) supports this format. This also provides a simple format for proprietary analysis models that wish to interface with SMS without a complete model specific interface.
- DXF: DXF is a standard file format specified by AutoDesk's AutoCAD®. Once the data from *SMS* is saved into DXF format, it may be read and edited by any program that supports this format

Exporting a TIFF image allows the user to a create an image file of the current SMS display screen. This image can then be imported to be used as background data, or edited digitally using an imaging software package.

#### 2.7.7 Get Info

The *Get Info* command brings up a dialog that reports basic information concerning the data type associated with the active module. For example, for meshes, the *Get Info* dialog reports the number of nodes, the number of elements, the number of linear elements, etc.

#### 2.7.8 Save Environment

The *Save Environment* used to save the current settings of the program (display options, defaults, etc.) to a default settings file. *SMS* reads the default settings file each time it is launched or the new command is invoked. This initializes the settings to the values stored in the file. If a file does not exist, SMS attempts to create one assigning default values.

#### 2.7.9 Print

Printed copies of *SMS* screen displays can be generated by using the *Print* command. The UNIX version of *SMS* will create a PostScript file that can be sent to a PostScript printer. The UNIX version will also create encapsulated PostScript files that can be imported into many other programs. The MS-Windows version will print to any printer supported by Windows.

If the *Plot Window* is up, the user is prompted to select a window to be printed. (see Figure 2.4

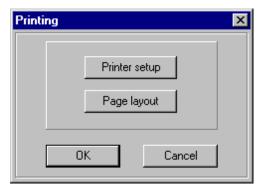


Figure 2.4 Page Select Dialog

When the *Print* command is selected, the *Print* dialog will appear (see Figure 2.5). This dialog allows the user to change a number of printing parameters. The *Printer Setup (PC)/Postscript Setup (UNIX)* button accesses the Windows *Printer Setup* 

dialog in the MS-Windows version and the *Page Size* dialog in the UNIX version. The *Page Layout* button accesses the *Page Layout* dialog. The data that will be printed to the output device is identical to that which appears in the selected window when a *Refresh* command is issued.

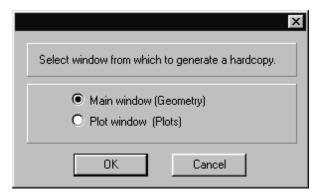


Figure 2.5 The Print Dialog.

#### **Printer/Postscript Setup**

The *Printer/Postscript Setup* command allows the user to control the orientation of the printed image and the paper size in UNIX (see Figure 2.6). The MS-Windows version brings up the printer setup dialog which allows the user to change relevant parameters of the currently selected printer. The current printer can also be changed using Windows Print Manager.

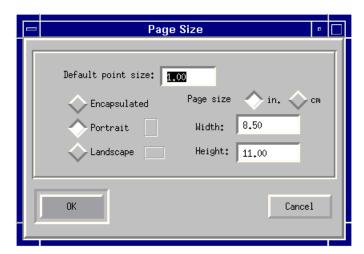


Figure 2.6 The Page Size Dialog.

#### Page Layout

The *Page Layout* dialog (see Figure 2.7) allows the user to change the size and position of the printed image on the paper. The image size is controlled by the two scroll bars just under the page display. When the *Maintain Aspect Ratio* box is checked, moving one of the scroll bars will also move the other scroll bar. The current image size is displayed to the right of each scroll bar. The *Center* buttons

allow the user to center the image on the page. The Max Aspect button sets the image to a size that will just fill the paper, maintaining the aspect ratio, with a 0.25 inch margin on either the left/right or top/bottom borders (depending on the restraining paper direction and paper orientation)

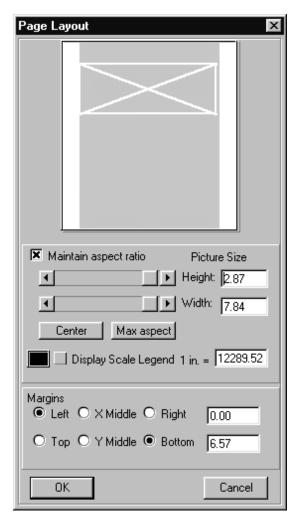


Figure 2.7 The Page Layout Dialog.

The top portion of this dialog displays how the image in the graphics window will be positioned on the printed page. The grayed region represents the paper, and its size and orientation will reflect the size and orientation specified in the Printer/Postscript Setup dialog.

The image may be interactively positioned on the paper by clicking on the box representing the image and dragging it within the paper display. In addition, the size of the image can be edited using the scroll bars and text edit fields in the center of the dialog, and the position can be edited using the margin control on the bottom of the dialog.

In the center of the dialog the user can control the scaling of the image onto the page. The size of the image is displayed in the same units as that used for the paper size and is displayed in the text edit fields on the right side. The toggle box at the top of this section can be selected to force the aspect ratio of the image to remain the same.

At the bottom of the dialog, the user can change the margins of the page. The horizontal margins can be referenced from the left, center or right. The size of the image controls the non specified values. For example, by specifying an image to be five inches wide, with a left margin of one inch on standard paper which is eight and one half inches wide, the right margin is left at two and one half inches, and the center point is at three and one half inches from the left. The vertical margins function similarly allowing the user to specify either the top, bottom, or center location.

#### 2.7.10 Demo Mode

Since some users may not require all of the modules or model interfaces provided in *SMS*, modules and model interfaces can be licensed individually. The icons for the unlicensed modules or the menus for model interfaces are dimmed and cannot be accessed. Even though you may have only licensed a portion of the *SMS* interface, the *Demo Mode* command provides a way of evaluating additional modules you may wish to consider licensing in the future. This is particularly useful when using the Tutorials, provided with this manual.

When the *Demo Mode* command is selected, all modules of the program will be enabled. The only exceptions are that the *Print*, *Save*, and *Export* options will be disabled. It is important to note that when the mode is changed all current data will be deleted. When the program is in demo mode, this menu command will toggle to read, *Normal Mode*. To return to normal operating mode, select the *Normal Mode* command. If an evaluation copy of the software is being executed, this menu item is unavailable.

## 2.7.11 Register

The *Register* command brings up a dialog which reports what modules are currently registered. For information on how to register your copy of *SMS* or enable additional modules consult your *SMS* installation guide or distributor.

The Details button allows the user to clearly read the registration string needed to obtain a password. Clicking on this button will open a dialog that gives a name to each letter and symbol. Using these names facilitates the issuing of a password.

#### 2.7.12 Quit

The *Quit* command is used to exit the program.

#### 2.8 Edit Menu

The *Edit* menu is one of the standard menus and is available in all of the modules. The commands in the *Edit* menu are used to select objects, delete objects, and set basic object and material attributes.

## 2.8.1 Delete

The *Delete* command is used to delete the selected objects. This command is equivalent to hitting the *DELETE* or *BACKSPACE* key.

#### 2.8.2 Select Commands

#### Select All

The Select All command selects all items associated with the current selection tool.

#### **Select With Poly**

The *Select With Poly* command allows the user to enter an irregular polygon enclosing the items to be selected (one of the selection tools must be active). To enter the polygon, locate the polygon's initial starting point and press down the right mouse button; continue in the same manner for each intermediate point defining the polygon and double click on the ending point. All items within the polygon will be selected.

#### **Select By Material**

Items may be selected as a group by using the *Select By Material* command. This command enables all nodes or elements which reference a specific material to be selected together. The application of materials is described below.

#### 2.8.3 Confirm Deletions

Whenever a set of selected objects is about to be deleted, the user is prompted to confirm the deletion. This is meant to ensure that objects are not deleted accidentally. This option can be turned off by selecting the *Do Not Confirm Deletions* command in the *Edit* menu. This menu command then changes to a *Confirm Deletions* command which can be used to turn the confirmation option back on.

#### 2.8.4 Materials

Many of the data types supported by *SMS* (i.e., elements, cells) have a material ID associated with each object. This material ID is an index into a list of material types. These material types often represent different types of bed material or areas of fluid

properties. A global list of material attributes is maintained that can be edited using the *Materials Data* command in the *Edit* menu. The command brings up the *Materials Editor* (see Figure 2.8) dialog where each material is assigned an ID number. This dialog can be used to delete unused materials, create new materials, and assign a descriptive name or color to a material. This general information is saved in the material file which is described in Sections 2.8.4 and 14.16.

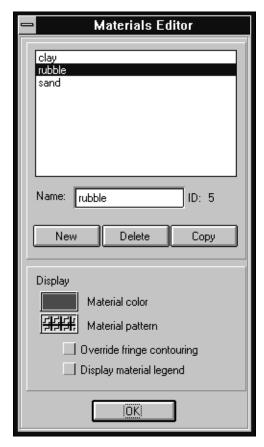


Figure 2.8 The Material Editor Dialog.

Only the general information concerning a material is edited with the *Materials* command. This general information includes the material name, color, and fill pattern. The user can specify if a material pattern should be displayed instead of contouring the element or cell by choosing the *Override fringe contouring* option. The *Display material legend* option places a legend of the materials in the *Graphics Window*. Model specific material properties such as Manning's n and Eddy viscosities are edited using commands local to the model menu.

When a new mesh or grid is created, a default material is assigned to the new object. Changing the default material is described in Section 4.8.

## 2.9 Display Menu

The *Display* menu is one of the standard menus and is available in all of the modules. The commands in the *Display* menu are used to control what entities are displayed, and the attributes of those entities. It also allows the user to control the display of a drawing grid, background color, and the range of values mapped to the display.

## 2.9.1 Refresh

When editing the image in the *Graphics Window* it occasionally becomes necessary to update the display or refresh the screen by redrawing the image. If desired, the user can tell SMS to automatically update the display when it detects it is required. This is done with the auto redraw option (section 2.9.2). If the display needs to be updated, it can be refreshed by selecting the *Refresh* command from the *Display* menu or by selecting the *Refresh* macro. The process of redrawing can be aborted by pressing the *ESC* key.

#### 2.9.2 Automatic Refresh

Depending on the capabilities of the computer being used, if a large model is currently in memory, significant time can be taken for the *Graphics Window* to refresh after making changes to the model. While at times it is useful to view the changes immediately after making them, it can sometimes become very tedious to have to wait for each redraw. The *Automatic Refresh* toggle provides the option to only refresh the *Graphics Window* when the *Refresh* command is issued.

If the *Automatic Refresh* command currently has an asterix displayed in the *Display* menu, selecting the command will toggle the refresh mode to manual. No further automatic updates to the *Graphics Window* are made until the *Refresh* command in the display menu (or the *Refresh* macro) is selected. If the image currently displayed in the *Graphics Window* is not up to date, the *Refresh* macro is highlighted in red. Immediately after issuing the *Refresh* command, the macro reverts back to normal display.

If the *Automatic Refresh* command currently has no asterix displayed in the *Display* menu, selecting the command toggles the refresh mode back to automatic.

## 2.9.3 Display Options

Each data type in SMS has a set of display options that can be modified using the Display Options command in the Display menu. The Display Options command brings up a different dialog for each module that can be used to control which features of the data type are to be displayed. For example, for the Scatter Point Module Display Options dialog, the user can choose to display scatter point labels,

and select a symbol to use for point display. The specific display options for each module are described in the appropriate chapter. Additional display options are specific to individual numerical models. These options are accessed through the numerical model interface menu, and described below in the chapter which discussing that interface.

Each display feature associated with a data type is listed in the *Display Options* dialog. The check box next to the feature named can be toggled on or off to control whether or not the feature is to be displayed. In addition, the attribute button to the left of the check box can be used to set the color of the feature.

## 2.9.4 Frame Image

After altering the image display using the *Zoom* or *Pan* tools, the image can be centered by selecting the *Frame Image* command in the *Display* menu. This command adjusts the window boundaries so that all currently visible objects just fit in the *Graphics Window*.

#### 2.9.5 Set Window Bounds

The region of the real world coordinate system that is mapped to the *Graphics Window* can be altered using the *Pan* and *Zoom* tools. It is also possible to precisely control the visible region by selecting the *Set Wind Bounds* command from the *Display* menu. The *Set Window Boundaries* dialog box appears, and the x and y limits of the viewing area can be set.

## 2.9.6 Grid Options

When entering new nodes or entering a polygon, it may be useful to have the coordinates snap to a uniform grid. This allows accurate placement of the objects when the desired coordinates are even multiples of some number.

A drawing grid can be activated using the *Grid Opts* command in the *Display* menu. If the *Snap to grid* option is selected, all new vertices will snap to the closest grid point. The grid spacing and options for displaying the grid can also be set using the *Drawing Grid Options* dialog.

## 2.9.7 Background Color

The user may select any of the standard twenty *SMS* colors as the background color. By selecting the background color item, a palette of the standard colors appears, from which the user specifies the color to be used for the *Graphics Window* background.

### 2.9.8 Plot Window

The user may bring into view or hide the plot window. Selecting the show plot window option will open the plot window and map it onto the display. Selecting the hide plot window option will cause the plot window to close. Time history gage plots (observation), cross section plots and profile plots are allowed only when the plot window is active.

#### 2.9.9 River Window

The user may bring into view or hide the river window. Selecting the show river window option will open the river window and map it onto the display. Selecting the hide river window option will cause the river window to close.

CHAPTER 3

# Data Visualization

*SMS* was designed as a general purpose modeling system. One of the main purposes of *SMS* is to provide a consistent interface for a variety of models and grid types. In order to accomplish this goal, input data and solution data are handled in a simple, consistent fashion using data sets.

A data set is a set of values associated with each node, cell, vertex, or scatter point in an object. A data set can be steady-state (one value per item) or dynamic (one value per item per time step). The values in the data set can be scalar values or vector values. Certain types of objects in *SMS* have an associated list of scalar data sets and a list of vector data sets. In *SMS*, both meshes and scatter points have a pair of data set lists. The commands for manipulating data sets are located in the *Data* menu. The *Data* menu is one of the standard menus and is available in each of the modules.

Data sets are used for both pre- and post-processing of models. For example, a scalar data set associated with a 2D mesh can represent starting or initial values of water surface elevation. *SMS* can export these values as an initial condition file for *FESWMS*. Another data set associated with the same mesh may represent computed velocity values. All data sets can be used to generate contours, color fringes, vector plots, and animation sequences. A detailed discussion of the visualization tools is presented in this chapter.

One advantage of the data set list approach for managing information is that it facilitates transfer of information between different types of models or models with differing resolution. This is accomplished through scatter point sets and interpolation. Meshes and grids can be converted to 2D scatter point sets. When an object is converted to a scatter point set, all data sets associated with the object are copied to the new scatter point set. The data sets can then be interpolated from the scatter point set to other objects of any type using one of the supported interpolation schemes.

## 3.1 A Browser

Most of the interaction with data sets is accomplished with the *Data Set Browser* (see Figure 3.1). The *Data Set Browser* is activated by selecting the *Data Browser* command in the *Data* menu. The two parallel list boxes in the browser contain the lists of scalar and vector data sets for the current object. In the case of the *Scatter Point Module*, the data sets shown in the browser correspond to the active scatter point set. In the case of meshes and grids, there is only one mesh or grid per module and the data sets correspond to the mesh or grid associated with the current module.

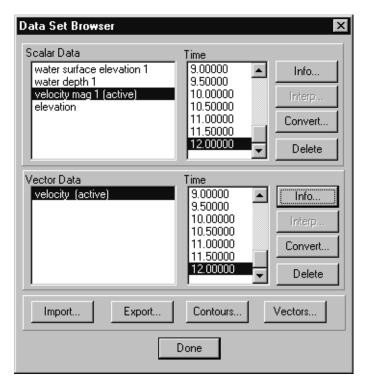
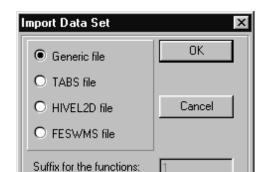


Figure 3.1 Data Browser Dialog.

#### 3.1.1 File I/O

Previously defined data sets can be input to *SMS* by selecting the *Import* button in the *Data Browser*. This will bring up the import type dialog box (see Figure 3.2). The user then selects the file type desired, from the supported types:

- Generic SMS ASCII Data Set Files
- Generic SMS Binary Data Set Files
- TABS (RMA2/RMA4/SED2D-WES) Solution Files
- HIVEL2D Solution Files



#### FESWMS Solution Files

Figure 3.2 Import File Types.

The *RMA2*, *FESWMS* files and HIVEL2D files are only accessible in the *Mesh Module*. Additional formats will be added as new computational models are supported. Generic *SMS* data set files can be imported in any module. The format for the *SMS* data set files is described in Chapter 14.

Generic files include specified function names. For other file types the user may input a suffix to be added to default function names. This will help in locating the desired data set within the file browser.

Once one of the file type options has been chosen, a file browser dialog appears and the user must select a file corresponding to the type selected.

Data sets can be exported from *SMS* to files by selecting the *Export* button in the *Data Browser*. Data sets can be saved as either binary or ASCII data set files in the *Data Browser* or from the save command (see section 2.7.4).

When a data set is imported to *SMS*, a copy of the data set is written to a temporary file on disk in binary form. If the imported data set is already in the form of a *SMS* binary data set file, a copy of the file is not made. When part of the data set is needed it is loaded from the hard disk into internal memory. Only one time step of one scalar data set and one vector data set is read into internal memory at any given time. This method of file manipulation reduces the amount of RAM required but it requires extra hard disk space. It also requires that a temporary location be defined for the system on which *SMS* is running.

When a new data set is created through interpolation or using the data calculator, a temporary binary file is created for the data set. To save the data set to disk permanently, the user must select the *Export* button from the *Data Browser*, or save the data from the *File* menu.

#### 3.1.2 Active Data Set

One data set is always highlighted in both the scalar and vector data set lists. In addition, if a dynamic data set is highlighted, the time steps for the data set are listed in the text box directly to the right of the list of data sets. One of the time steps is highlighted. The highlighted sets are the active data sets for the object. The values corresponding to the active data sets and time steps are used whenever contour, color fringe, and vector plots are generated. In addition, the entire range of time steps of the active data sets are used whenever animation film loops are generated.

Whenever a new data set is created by importing from a file, interpolating, or using the data calculator, the data set becomes the active data set for the object.

#### 3.1.3 Elevations & Automatic Data Sets

Whenever a new mesh object is created or read from a file, a scalar data set is created containing the elevations of the nodes of the object. Thus, there is always at least one data set associated with a mesh. This data set cannot be deleted.

There are several other situations which SMS will automatically create data sets. These include SED2D-WES bed property sets (see section 9.2.1) and RMA10 layer data sets.

### 3.1.4 Deleting Data Sets

Data sets can be deleted by selecting the data set in the list box and selecting the Delete button in the *Data Browser*. This removes the data set from the list and deletes the binary copy of the data set on disk. If the original data set file was already in binary form, the file is not deleted.

All data sets associated with an object are automatically deleted whenever the object is deleted or whenever the number of nodes or scatter points in the object is changed due to an editing command.

#### 3.1.5 Data Set Info

The *Info* buttons in the *Data Browser* will bring up a dialog listing some of the main characteristics of the active vector or scalar data set. These characteristics include statistics such as maximum, minimum, and range as well as mean and standard deviation The name of the active data set can also be edited from the *Info* dialog.

### 3.2 Data Calculator

The *Data Calculator* can be used to perform mathematical operations with data sets (see Figure 3.3). The *Data Calculator* can be accessed by selecting the *Data Calculator* command from the *Data* menu.

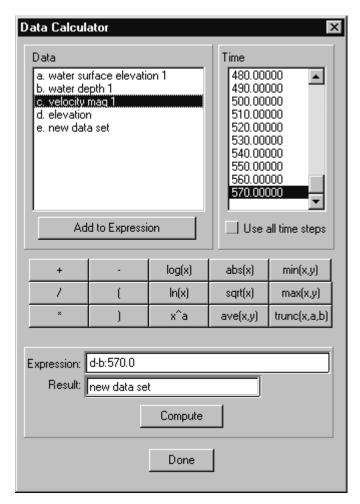


Figure 3.3 The Data Calculator.

The *Data Calculator* can be used to perform any set of mathematical operations shown as icons in the center of the dialog. Some of the operators are binary (i.e., "+", "-") and some are unary (i.e., "1/x",  $\ln(x)$ ). The user simply composes a mathematical equation using operators and data sets. The operator then specifies a name for the new data set to be created as a result of the operation. Once the mathematical operation is defined, the user clicks on the COMPUTE button to execute the operation. The new function then appears in the *Data Set* list.

The *Data Calculator* is useful for a variety of tasks. For example, to generate a data set representing the absolute difference between two other data sets, the user enters the equation |a-b| where 'a' and 'b' correspond to the two data sets. Such a data set

would be useful for comparing the results of two separate solutions computed by a numerical model.

# 3.3 Wector Options

If the *Vectors* item in the *Display Options* dialog is toggled for display, vector plots can be generated using the active vector data set for the object. One vector is placed at each node or cell. The display of vectors can be controlled using the *Vector Options* dialog (see Figure 3.4) accessed through the *Vector Opts* command in the *Data* menu or from the *Data Browser* or *Display Options* dialogs.

The upper portion of the *Vector Options* dialog allows the user to specify the shape of the vector arrows by specifying proportional head length, head width, shaft length and shaft width.

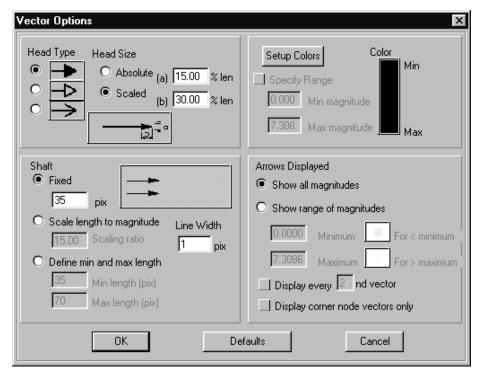


Figure 3.4 Vector Options Dialog.

In the lower left, the user specifies how big the arrows will appear in the graphics window. This can be a constant length, a scaled length, or a range of lengths. The min and max arrow size are displayed in the canvas just above this section of the dialog.

The lower right portion of the dialog is used to specify the arrow color and density of arrow to be displayed. In the top part of this section of the dialog, the user specifies whether all arrows should be a constant color, or a ramp of colors. If a ramp of colors is used, the color ramp described below provides the colors for arrow display. By

default, the arrow with the smallest magnitude is displayed in the color at the bottom end of the ramp, and the arrow with the largest magnitude is displayed in the color at the top of the ramp. Intermediate magnitudes are interpolated to intermediate colors. Alternately, the user can define the magnitudes that map to the top and bottom of the ramp. If this option is used, any arrow with a magnitude lower than the minimum is displayed in the color at the bottom of the ramp, and any arrow with a magnitude greater than the maximum is displayed with the color at the top of the ramp. Arrows with magnitudes between the minimum and the maximum are displayed in an interpolated color from the ramp. In the bottom section of the lower right portion of the dialog the user can reduce the number of vectors displayed. In a very dense mesh, a large number of data points may be displayed very close together on the screen. Therefore, if a vector is displayed at every point, the picture can become a jumble of vectors on top of each other. One way to treat this is to zoom in on a specific portion of the mesh, so the nodes are not displayed so close together. However, if the desired region of the mesh is still to dense, the user can turn off the vectors at the midside nodes, or display only a portion of the vectors. The user may also choose to only display arrows in a certain range. This may be used to turn off arrows of insignificant length, as well as those so large they occlude important details. Generally, a balance can be obtained so that enough vectors are displayed to get a feel for the flow field, and not be too cluttered.

The default values button at the bottom fills in a set of values based on the values in the initial conditions file.

#### 

Elements and grids can be contoured by turning on the contour option in the *Display Options* dialog (see Figure 3.5). When an object is contoured, the scalar values associated with the active data set for the object are used to generate the contours.

The options used to generate contours can be edited by selecting the *Contour Options* command in the *Data* menu or by selecting the *Contour Options* button in either the *Data Browser* or *Display Options* dialogs. The *Contour Options* dialog is shown in Figure 3.5. The values shown in the upper left corner of the dialog correspond to the maximum and minimum values in the active data set. These values are sometimes useful when choosing an appropriate contour interval.

The contour interval can be specified either by specifying a contour interval, a total number of contours (from which the contour interval is computed), or a set of explicit contour values. For either a specified interval or specified number of contours options, a maximum and a minimum contour value can be specified and the contouring can be restricted to this specified range. If no explicit range is specified, *SMS* will choose a range to optimize the color distribution inside the extreme values for the data set. If the *Values* button is selected, the *Contour Values* dialog is displayed. Up to ten specific contour values can be typed into the dialog.

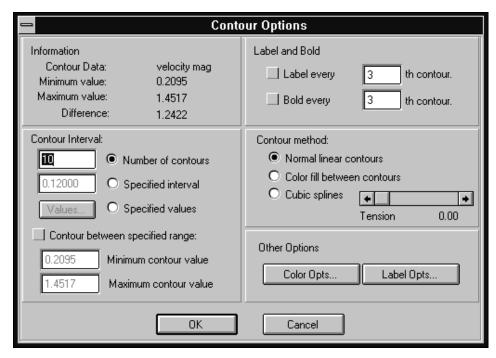


Figure 3.5 Contour Options Dialog.

The items in the upper right section of the *Contour Options* dialog are used to control the graphical appearance of the contours. Contours at selected intervals can be automatically labeled and/or displayed with a thicker line width.

The options in the middle of the right side of the dialog control how the contours are computed. Three contouring methods are available:

- The default method is *Normal Linear Contours* and causes the contours to be displayed as piece-wise linear strings.
- If using the *Color fill between contours* method, the region between adjacent contour lines is filled with a solid color.
- If using the *Cubic Spline Contours* method, the contours are computed in strings and drawn as cubic splines. Drawing the contours as splines can cause the contours to appear smoother. Occasionally, loops appear in the splines or the splines cross neighboring contour splines. These problems can sometimes be fixed by adding tension to the splines. A tension factor greater than zero causes the cubic spline to be blended with or converge to a linear spline based on the same set of points. A tension factor of unity causes the cubic spline to coincide with the linear spline.

In the lower right corner of the *Contour Options* dialog, two buttons allow the user to specify the contour color and the contour labeling options.

#### 

The *Color Ramp Options* dialog (see Figure 3.6) lets the user determine how the contours, and vectors will be colored. The user may invoke the *Contour Color* dialog either from the *Data* menu, or from a button in the *Data Browser* or *Contour Options* dialogs. The default contour color method is the *Use default contour color* option. This method allows the user to select a color from the 20 standard *SMS* colors and all contours are displayed with that color. As an alternative, the user can define a ramp of colors. These colors are distributed across the range of contour values in a continuous fashion, giving each contour its own color. There are two types of ramps supported by *SMS*.

- If using the *Vary default contour color brightness* option, the color ramp will be defined as a continuous variation of the intensity of the default contour color. This is the same color used for the *Use default contour color* option.
- If using the *Range of hues* option is chosen, the ramp will be defined as a continuous variation of hues using the hue-saturation-value color model.

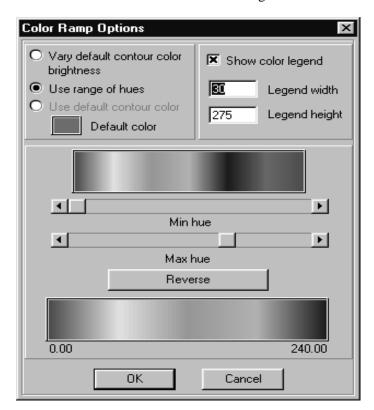


Figure 3.6 Color Ramp Options Dialog.

In either case, the minimum and maximum ramp color can be altered by the user using the horizontal scroll bars in the *Contour Color Options* dialog. The reverse button changes the direction of the color gradation in the color ramp

If the *Show color legend* option is selected, and the color option is one of the ramps, a legend of colors and corresponding data set values is displayed in the upper left corner of *Graphics Window*. This legend is a vertical strip of colors with text labels for the contour levels. If the contours are being displayed as linear segments or cubic splines, the legend is displayed as a series of contour level values and a line drawn in the color corresponding to that level. If the *Use default contour color* option is selected, the *Show color legend* toggle is ignored.

#### 

The *Contour Label Opts* command in the *Data* menu is used to access the *Contour Labels Options* dialog which can be used to set the label color, label size, etc. The dialog may also be invoked through the *Contour Options* dialog. The default spacing value controls the placement of labels when labels are generated automatically.

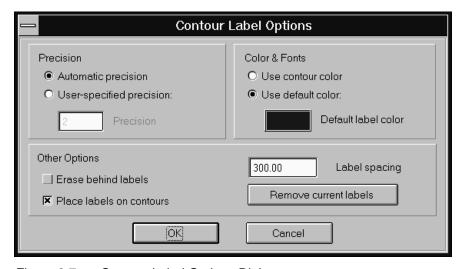


Figure 3.7 Contour Label Options Dialog.

Labels can be added to contours one of two ways:

- If the label contours toggle is selected in the *Contour Options* dialog, labels are automatically placed on the appropriate contours.
- In some modules, contour labels can be added manually to contours by selecting the *Contour Labels* tool in the *Tool Palette* and clicking on the contours where labels are desired. By default, the data set value corresponding to the point that was clicked is computed and a label corresponding to the nearest contour value is drawn centered at the point that was clicked. If the *Place labels on contours* option in the lower left portion of the *Contour Label Options* dialog is on, the label is moved to the closest point on the contour closest to the selected point. The value of the label is the value of the contour. If this option is not selected, the label shows the value of the point at the click location and is placed there. This option is useful to

post data set value labels in regions where there are no contours. If the mouse button is held down, a box showing the outline of the label is drawn. The box can then be positioned precisely with the mouse. A line is drawn from the box to the point that was clicked to help the user keep track of the contour that was selected. Contour labels can be deleted by holding down the *SHIFT* key while clicking on a label.

## 3.7 Film Loops

One of the most powerful visualization tools in *SMS* is animation. An animation sequence can be generated for an object with a dynamic data set to illustrate how vectors, contours, fringes, or iso-surfaces vary as a function of time. Each frame of the animation may be stored as a pixel map. The entire set of frames in an animation sequence is referred to as a film loop.

Animation film loops are generated by selecting the *Film Loop* command in the *Data* menu. This command brings up the *Film Loop* dialog shown in Figure 3.8. The *Film Loop* dialog is used to control the film loops. A new film loop can be generated by selecting the *Setup* button. Once a film loop has been generated, it can be saved to a file using the *Save* button. Previously saved film loops can be read from disk using the *Read* button.

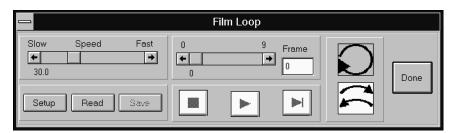


Figure 3.8 The Film Loop Dialog.

### 3.7.1 Film Loop Setup

A new film loop can be generated by selecting the *Setup* button in the *Film Loop* dialog. This button accesses the *Film Loop Options* dialog (see Figure 3.9).

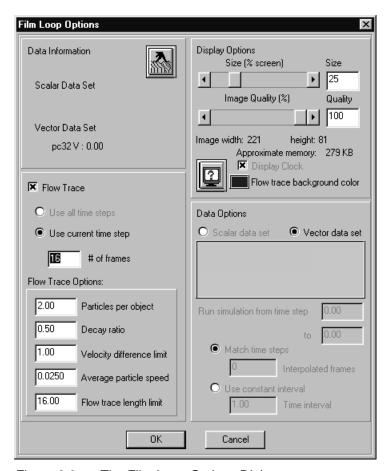


Figure 3.9 The Film Loop Options Dialog.

#### **Data Set**

Film loops are always generated using the active data set. The *Data Browser* button in the Data Information portion of the dialog can be used to change the active scalar and vector data sets. The current active data sets are displayed to the left of the *Data Browser* button.

#### **Image Size**

Each frame that is generated in a film loop occupies a portion of the entire *Graphics Window*, as specified by the scrollbar in the Display Options portion of the dialog. Naturally, the most pleasing animations are comprised of full screen images. However, this may result in film loops which require a significant amount of memory and which are difficult to playback at a high speed. By reducing the frame size, the film loop itself becomes both smaller and faster. The memory required for a film loop is quadraticly proportional to the fractional size. For example, an image generated at 50% of the *Graphics Window* size requires 25% as much memory as an image generated at full *Graphics Window* size.

#### **Animation Time Control**

Animation can be applied to any object with a dynamic data set. As each frame is generated, a set of values corresponding to the current time is loaded into memory and the image is redrawn using the current display options. Thus, if the contour display option is selected, the contours will vary from frame to frame. If the vector option is selected, the vectors will vary from frame to frame.

The strip in the center of the Data Options portion of the *Film Loop Setup* dialog is used to specify what range of the available time steps are to be used for animation. The range of time steps can also be entered directly in the edit fields below the time step strip. The range displayed in the strip corresponds to either the scalar or vector data set depending on the status of the radio group above the time step strip. *SMS* can only animate one type of data at a time. Therefore only the selected type of data set is used for generation of the film loop or animation.

The total number of frames generated in the film loop can be defined by either matching the time steps (one frame per time step) or by using a constant interval (e.g., one frame for every two hour interval). If the *Match Time Steps* option is chosen, extra frames can be created between each time step using linear interpolation of the data values at the specified time steps.

#### Flow Trace Animation

Flow trace animation is a technique used to visualize vector fields in *SMS*. It can be thought of as dropping tiny drops of dye into a fluid field in a random distribution and watching the flow pattern created. The process can also be thought of as creating massless particles and letting the vectors in the vector field be forces pushing the particles around. The Flow Trace portion of the *Film Loop Setup* dialog allows the user to control the flow trace. This entire portion of the dialog is disabled if no vector data exists for the current data set. The top radio group allows the user to specify whether the flow trace should be created for a steady-state or dynamic system. Below this the user can specify the density of particles or dye droplets by specifying the average number of particles for each cell or element. The number of frames required for a droplet to become dispersed is represented as a portion of the animation in the *Decay ratio* field.

The path of each particle is defined by tracing the particle from a starting position to successive points through the field. The next point is computed by moving the particle from the previous point according to the value of the vector field at that previous point. At the new point, the velocity and direction are sampled. If the particle has traveled farther than the *Flow trace length limit*, or the velocity has changed more than the *Velocity difference limit*, the step is broken into two steps of half the step size. This process is repeated, until a sequence of valid points within the limits are defined for each frame. Therefore, the smaller the values of the *Flow trace length limit* and *Velocity difference limit*, the more precisely the particles will imitate the vector field. Generally, the default values are sufficient.

The Average particle speed is used to scale the vector field, thus changing the distance each particle or droplet travels. This is useful for vector fields with extreme magnitudes. For a low magnitude data set, the particles may not move very far. While this sluggish motion is accurate for the data, scaling the vector field up, and exaggerating the motion causes the flow patterns to be more visible. Similarly, in high magnitude fields the particles may become long streaks and scaling the values down may result in a clearer picture of the flow patterns.

### 3.7.2 Film Loop Playback

Once a new film loop has been generated or a film loop has been read from disk, several options are available for playing back the film loop. The buttons at the lower middle of the *Film Loop* dialog are designed to mimic the buttons on a VCR or CD player. The *Play* button causes the film loop to cycle continuously. The *Stop* button halts the playback. The *Step* button can be used to advance the film loop forward one frame at a time. In addition, the frame scroll bar can be used to interactively move the frames forward or backward.

The speed of playback can be adjusted using the *Speed* scroll bar. The maximum speed depends on the speed of the computer and the size of the image being animated. The smaller the image, the faster the maximum playback speed.

Two options are available for cycling the film loop playback. The *continuous* playback option starts a new cycle at the first frame in the loop after the last frame is encountered. The *oscillation* option plays the loop in the forward direction to the end of the loop and then in the reverse direction back to the beginning of the loop.

### 3.7.3 Saving Film Loops

Saving and reading film loops is useful since some film loops may take a significant amount of time to generate depending on the complexity of the image. The film loops are saved to disk in a compressed binary format. When a film loop is read from disk, it is first uncompressed on the hard disk. Then, if sufficient internal memory is available, the entire film loop is read into RAM for playback. If sufficient internal memory is not available, the playback is performed reading one frame at a time from the hard disk.

With the PC version of *SMS*, film loops are saved in the *MS Video for Windows* (\*.AVI) format. AVI files can be embedded into Presentations and PowerPoint presentations and other multi-media documents. They may also be viewed using any stand alone AVI video for windows player. This offers the ability to present and review film loops outside of the SMS environment.

### 3.8 Gages

One of the most important steps in any modeling problem is verification. During the verification phase, an attempt is made to model a set of conditions which have been known to exist at a site and for which measured data (water surface elevations, concentrations) are available. The geometry, resolution, and input parameters of the model are adjusted until the output computed by the model is reasonably close to the measured data.

The verification stage can be the most tedious and time-consuming portion of the modeling process. In order to make the verification stage more efficient, a set of tools for managing gages has been provided in *SMS*. A "gage" is an XY location defined by the user representing a location where field data has been collected (i.e., a flow meter), or simply a point of interest in the model. Once a set of gages has been defined, whenever a dynamic data set is imported into an *SMS* mesh or a grid, the data set is interpolated to each gage and a curve is drawn in the plot window representing the variation of the data set with time at each gage. The plot can be customized to include any combination of gages and data sets. Field data can be imported from text files and plotted with computed curves for comparison.

Gages and gage plots are supported in the *Mesh Module* and in the *Grid Module*. One set of gages can apply to all modules. In other words, if both a finite element mesh and a finite difference grid are in memory, data sets from both meshes can be interpolated to the same set of gages. There is no need to define a separate set of gages for each module.

### 3.8.1 The Gages Dialog

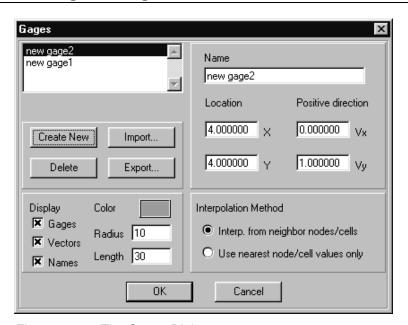


Figure 3.10 The Gages Dialog.

A gage or set of gages can be created by selecting the *Gages* command from the *Data* menu. This command displays the *Gages* dialog as shown in Figure 3.10. All existing gages are listed in the text box in the upper left corner of the dialog. One of these gages is highlighted at all times. The name, location, and direction vector of the highlighted gage can be edited using the controls on the right hand portion of the dialog.

The color and name are used in displaying the gage in the *Graphics Window*. The positive direction vector is used for plotting curves extracted from vector data sets. When computing a curve from a vector data set, the magnitude of the interpolated vector is plotted. The sign (+ or -) of the vector depends on the direction of the vector relative to the positive direction vector. If the vector is in the same halfspace as the positive direction vector (the dot product is positive), then the magnitude is plotted as a positive number. If the vector is in the opposite halfspace as the positive direction vector (the dot product is negative), then the magnitude is plotted as a negative value. The positive direction vector for a gage may be edited using the text edit fields of the *Gages* dialog, or the user may select the *Select Gage* tool from the *Tool Palette* and click and drag the arrow head in the *Graphics Window*.

A new gage can be added to the list of gages by selecting the *Create New* button beneath the list of gages. The highlighted gage can be removed from the list by selecting the *Delete* button. A set of gages can be imported from a text file by selecting the *Import* button. The format of the gage file is described in Chapter 14. A set of measured curves can be included in the gage file for comparison with computed curves. A set of gages created within *SMS* can be exported to a file for future use by selecting the *Export* button.

The *Interpolation Method* options in the lower right corner of the *Gages* dialog controls how data sets are interpolated to the gages for curve plotting. If the *Interp. from neighboring nodes/cells* option is chosen, the data sets are interpolated from the nodes or cells in the vicinity of the gage using a simple inverse distance weighted interpolation scheme. If the *Use nearest node/cell values only* option is selected, interpolation is not performed. The node or cell closest to the gage is found, and the data set values at that node or cell are then used to generate the curve for the gage.

Gages are plotted in the *Graphics Window* as shown in Figure 3.11. The arrows drawn on the gages represents the positive direction vector for that gage. The gage name is plotted just below the gage symbol. Each component of the gage can be turned on or off or resized using the *Display* items in the lower left corner of the *Gages* dialog.

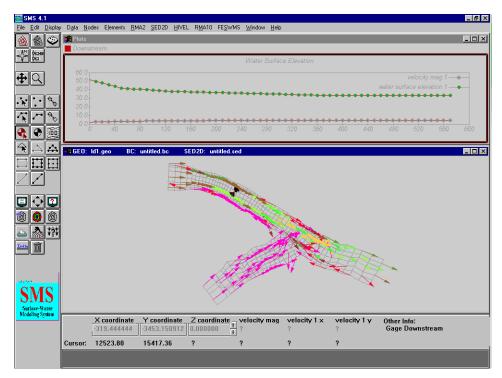


Figure 3.11 An Example of Gage Symbols in the Graphics Window.

### 3.8.2 The Gage Tools

Gages can also be created and deleted using the *Gage* tools. The gage tools appear in the dynamic portion of the *Tool Palette* of each of the modules which support gages. The tools are as follows:

## The Create Gages Tool

The *Create Gages* tool is used to interactively create gages in the *Graphics Window*. When this tool is active, a new gage is created by clicking in the *Graphics Window* at the desired location of the gage. The xy coordinates of the gage are defined by the cursor position. The x and y coordinates of a new gage can be edited using the *Edit* Window. In addition, once a gage has been defined with the *Create Gages* tool, the gage can be edited using the *Gages* dialog or edited graphically using the *Select Gages* tool. This is described in the next section.

# The Select Gages Tool

The *Select Gages* tool is used to select previously defined gages. A set of selected gages can be deleted by pressing the *DELETE* key or by selecting the *Delete* command from the *Edit* menu. The coordinates of a selected gage can be edited using the *Edit Window*. The location of a gage can also be edited by clicking on a gage, holding down the mouse button, and then dragging the gage with the mouse. If the click is outside of a gage, but close to it, dragging the mouse edits the positive

direction of the gage. This tool is also used to control what is plotted in the *Plot Window*. Only the curves associated with selected gages are plotted.

### 3.9 The Plot Manager

Once a set of gages have been defined, one or more plots can be generated in the *Plot Window* representing the variation vs. time of any of the dynamic data sets associated with grids or meshes interpolated to the gages. Up to five plots may be generated at once. Any combination of data sets can be displayed on a single plot. The curves are plotted only for gages which have been selected using the *Select Gages* tool. This makes it possible to quickly and easily change the combination of curves plotted.

The *Plot Manager* command in the *Data* menu activates the *Plot Manager* dialog as shown in Figure 3.12. The names of five plots are listed in the text box, whether they are displayed or not. The word *visible* or *hidden* will appear beside each plot title indicating whether the plot is currently displayed in the *Plot Window*. The titles of the plots are listed in the order that they are displayed. One of these plots is highlighted at all times. To display a plot in the *Plot Window*, first select the plot to be displayed in the text box, then select the *Make Visible* button. To hide the plot, select the *Hide Plot* button.

The *Gages* button in the *Plot Manager* brings up the *Gages* dialog described above. This dialog can also be activated using the *Gages* command in the *Data* menu.

The *Print* button in the *Plot Manager* prints a copy of the plots currently displayed in the *Plot Window*.

The *Export WKS* button exports a copy of the points used to create the plots in the plot window to a common tab delimited spreadsheet file. This allows more customization of the plots by an external spread sheet program, if necessary.

Currently displayed plots will be updated immediately as updates are made to gage locations and as gages are selected and deselected. The *Plot Window* may be closed by selecting the Hide Plot Window option in the edit menu. With the UNIX version, the *Plot Window* can also be closed by selecting the Close menu item from the pull-down menu in the upper left corner of the X-window.



Figure 3.12 The Plot Manager Dialog.

#### The Plot Curves Dialog

The set of curves displayed in each plot can be edited by selecting the *Curves* button in the *Plot Manager*. This button activates the *Plot Curves* dialog, as shown in Figure 3.13.

Two lists of curves are displayed in the dialog. One of the curves is always highlighted in each list. The list on the left represents the curves which are available for plotting. The list on the right represents the curves which are being displayed in the selected plot. Initially, the list on the right contains one curve for each data set. These curves represent the values of the data set interpolated to the gage locations. The list also contains one curve for each of the measured curves imported with a gage file. Curves in the available lists are moved to the list of plotted curves using the  $Selected \rightarrow All \rightarrow Button$  moves the highlighted curve and the  $All \rightarrow Button$  moves all of the curves. Likewise, curves can be moved from the plotted curves list to the available curves list using the  $Selected \rightarrow Button$ .

The display options for the highlighted curve in the plotted curves list can be edited using the group of controls at the bottom of the *Plot Curves* dialog. The curves can be plotted by displaying a symbol at each point on the curve or by displaying a line through the points or with a combination of points and lines. The symbol, line thickness, and line style can all be edited.

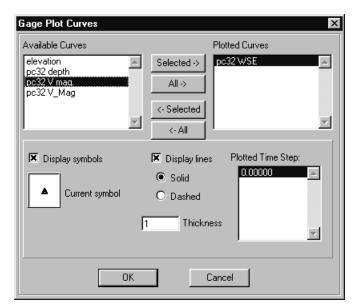


Figure 3.13 The Plot Curves Dialog.

### **The Plot Options Dialog**

The options used to display the plots listed in the *Plot Manager* can be edited using the *Plot Options* dialog. The options for an individual plot are edited by selecting the plot in the plot list and then selecting the *Plot Options* button. This activates the *Plot Options* dialog, as shown in Figure 3.14.

A major and a minor title can be entered. Both titles are displayed at the top of the plot with *Title 1* displayed above *Title 2*. The *X* and *Y Titles* are displayed along the x and y axes.

The *Foreground color* is used to display the titles, axes, grid lines, etc. The *Background color* is used to fill in the background of the plot. The *Curve legend* option causes a legend to be printed on the plot, showing the symbol and line style used for each curve next to the curve name.

The x and y axes can be displayed using either the *Autoscale* option or the *Manual scale* option. With the *Autoscale* option, the range, tick interval, label interval, and grid line interval are all chosen automatically. With the *Manual scale* option, each of these values can be specified explicitly.

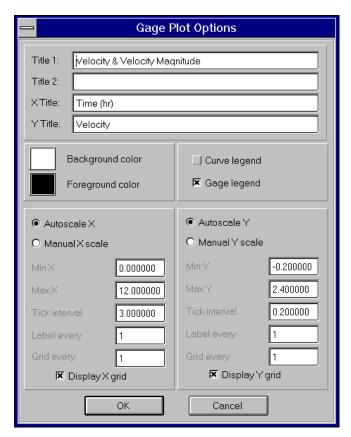


Figure 3.14 The Plot Options Dialog.

CHAPTER 4

# Mesh Module Mesh Module

The *Mesh Module* is used to construct and edit 2D finite element meshes. A mesh is defined as a network of triangles and quadrilateral elements constructed from nodes. A general set of tools are provided for automated mesh generation and mesh editing. These tools represent the automation of traditional mesh generation technologies. They can be used directly, or in conjunction with fully automated mesh generation techniques provided by the *Map Module* and discussed in chapter 6. Once a mesh is generated, boundary conditions and material properties for a specific numerical model such as *RMA2*, can be assigned using the commands in the menu associated with the model. The basic set of tools for mesh generation and editing are described in this chapter. Some of the tools previously available in this module have been superseded by automatic mesh generation tools included in the *Map Module*. These include the mesh from polygon capabilities that were included in the *element menu* in previous versions of *SMS*. The commands in the model menus are described in later chapters.

### 4.1 Tool Palette

The following tools are contained in the dynamic portion of the *Tool Palette* when the *Mesh Module* is active.

## 4.1.1 Create Mesh Nodes

The *Create Mesh Nodes* tool is used to manually add nodes to a mesh. When this tool is selected, clicking on a point within the *Graphics Window* will place a node at that

point. What happens to the node after it is added (whether and how it is triangulated into the mesh) depends on the settings in the *Node Options* dialog in the *Node* menu.

## 4.1.2 Select Mesh Nodes

The *Select Mesh Nodes* tool is used to select a set of nodes for some subsequent operation (such as deletion). The coordinates of a selected node can be changed by dragging the node while this tool is active (if the nodes are un-locked, see section 4.5.3). The coordinates of selected nodes can also be edited using the *Edit Window* (see section 2.5).

# 4.1.3 Create Nodestrings

The *Create Nodestrings* tool is used to create strings of nodes. Nodestrings are used for operations such as assigning boundary conditions, forcing breaklines into the mesh, or creating elements from patch boundaries. The procedure for creating nodestrings is as follows:

- 1. Click on the first node in the string. This node will be highlighted in red.
- Click on any subsequent nodes you would like to add to the string (nodes do not have to be next to each other). The selected nodes are connected by a solid red line.
- 3. Clicking a second time on the last node of a string, or pressing enter finishes the creation process, and the nodestring creation is complete. Another nodestring can then be created.
- 4. To remove the last node from a string, press the *BACKSPACE* key. To abort entering a nodestring, press the *ESC* key.

Nodestrings can also be automatically generated along a boundary line using the shift and control keys.

- 1. Click on the first node in the string. This node will be highlighted in red.
- 2. Holding down the *SHIFT* key and selecting another node will select nodes along element edges between the two nodes. The path will try to be straight, but may vary. SMS may be unable to find a path along element edges to the next node. If this happens, you will be asked how to proceed.
- 3. Holding down the *CTRL* key and selecting the next node will select nodes along the mesh boundary going counter clockwise to the selected node. Both the previous node and the selected node must be on the boundary of the mesh.

4. Holding down both the *CTRL* and *SHIFT* key and selecting the next node will select nodes along the mesh boundary going clockwise to the selected node. Both the previous node and the selected node must be on the boundary of the mesh.

## 4.1.4 Select Nodestrings

The *Select Nodestrings* tool is used to select one or more previously defined nodestrings for some subsequent operation such as assigning boundary conditions or patch creation. An icon for each nodestring appears as soon as this tool is selected. One nodestring is selected by clicking on its icon. If multiple nodestrings are desired, or required for the operation, subsequent selections may be made by clicking on additional icons while holding down the *SHIFT* key. Selections may also be made by dragging a box around one or more nodestrings icons. Selected nodestrings are displayed in red color, and their icons are filled.

## 4.1.5 Create Gages

The *Create Gages* tool is used to interactively create gages in the *Graphics Window*. When this tool is active, a new gage is created by clicking in the *Graphics Window* at the desired location of the gage. The xy coordinates of the gage are defined by the cursor position. The coordinates of a new gage can be edited using the *Edit Window*. In addition, once a gage has been defined with the *Create Gages* tool, the gage can be edited using the *Gages* dialog. Gages are described in detail in Chapter 3.

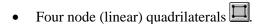
## 4.1.6 Select Gages

The Select Gages tool is used to select previously defined gages. Multiple gages can be selected by clicking and dragging or by holding down the SHIFT key while selecting the gages. A set of selected gages can be deleted by pressing the DELETE key or by selecting the Delete command from the Edit menu. The coordinates of a selected gage can be edited using the Edit Window. The location of a gage can also be edited by holding down the mouse button when a gage is selected and then dragging the gage. In addition to editing a gage's location, the positive direction of the gage can also be interactively edited by clicking on the arrow point and dragging it around the gage. This tool is also used to control what is plotted in the Plot Window. Only the curves associated with selected gages may be plotted.

#### 4.1.7 Create Elements

Six type of elements are supported by the *Mesh Module*:

Two node (linear) lines.



- Three node (quadratic) lines.
- Six node (quadratic) triangles
- Eight node (quadratic) quadrilaterals
- Nine node (quadratic) quadrilaterals with center node

Elements can be created using automatic meshing techniques such as triangulation. However, it is often necessary to edit a mesh by creating elements one at a time using one of the available *Create Element* tools.

A single element can be constructed from a set of existing nodes using the following steps:

- 1. Select the tool corresponding to the type of element to be created: two node line, three node line, three node triangle, six node triangle, four node quadrilateral, eight node quadrilateral or nine node quadrilateral.
- 2. Select the nodes corresponding to the <u>corner</u> nodes of the element in consecutive order around the perimeter of the element. The nodes can be selected in either clockwise or counter-clockwise order. It is also possible to build an element by dragging a rectangle to enclose the nodes making up the new element rather than selecting each node one by one. A beep will sound if the wrong number of nodes for the current element type have been selected.

If the current element type is a quadratic element (three node line, six node triangle, eight or nine node quadrilateral), the midside nodes of the element are created automatically. If the new element is adjacent to an existing element, the midside node of the existing element is used for the new element and a new midside node is not created (i.e., midside nodes are not duplicated). For nine node quadrilaterals, the center node is also created automatically when the element is created.

SMS performs several checks when a new element is constructed. The new element is checked to see whether or not it is ill-formed (i.e., the element has a twist in it or is self intersecting). The element is also checked to see if it overlaps any of the elements adjacent to the nodes comprising the new element. In addition, elements SMS does not allow linear elements such as three noded triangles and four noded quadrilaterals to exist in the same model with quadrilateral elements such as six noded triangles, eight and nine noded quadrilaterals. If any of the above checks fails, the construction of the new element is aborted.

## 4.1.8 Select Elements

The *Select Elements* tool is used to select a set of elements for some subsequent operation such as deletion, refinement, or assigning a material type.

## 4.1.9 Swap Edges

Occasionally, it is useful to interactively or manually swap the edges of two adjacent triangles. This can be thought of as a quick and simple alternative to adding breaklines (sec. 4.7.6) to ensure that the edges of the triangular elements honor a geometrical feature that needs to be preserved in the mesh.

If the *Swap Edges* tool in the *Tool Palette* is selected, clicking on the common edge of two adjacent triangles will cause the edge to be swapped, provided that the trapezoid formed by the two triangles is not concave.

## 4.1.10 Merge/Split

If the *Merge/Split* tool is selected, clicking on a triangle edge with the mouse cursor will cause two triangular elements adjacent to the edge to be merged into a quadrilateral element. They must be two elements adjacent to the edge, and they both must be triangular. In addition the trapezoid formed by the two triangles can not be concave.

The *Merge/Split* tool can also be used to undo a merge (i.e., unmerge) or split a quadrilateral element. A quadrilateral element can be split into two triangles by clicking anywhere in the interior of the element.

## 4.1.11 Wall Label Contours

The *Contour Label* tool manually places numerical contour elevation labels at points clicked on with the mouse. These labels remain on the screen until the contour options are changed, until they are deleted using the *Contour Labels* dialog, or until the mesh is edited. Individual contour labels can be deleted using this tool by holding down the *SHIFT* key while clicking on a label.

### 4.2 Mesh Conversion

It is sometimes useful to convert a finite element mesh to one of the other data types supported in *SMS*. Meshes can be converted to scatter point sets in the *Data* menu of the *Mesh Module*.

#### 4.2.1 Mesh -> Scatter Point

The Mesh -> Scatter Points command in the Mesh Data menu is used to create a new scatter point set using the nodes in a mesh. A copy is made of each of the data sets associated with the mesh and the data sets are associated with the new scatter point set.

This command is useful for comparing the solutions from separate simulations of the same geographical region. For example, if two simulations have been performed with slightly different meshes (i.e., base vs. plan) it may be useful to generate a contour plot showing the difference between the solutions. It is possible to generate a data set representing the difference between two data sets using the *Data Calculator* (see Figure 3.3). However, the two data sets must be associated with the same mesh before the *Data Calculator* can be used.

The data sets from one of the meshes can be transferred to the other mesh as follows:

- 1. Load the first mesh and its data set into memory.
- 2. Convert the mesh to a scatter point set using the *Mesh* -> *Scatter Points* command.
- 3. Load the second mesh (.geo file) and its data set into memory. This operation deletes the first mesh at the same time.
- 4. Switch to the *Scatter Point Module* and select an interpolation scheme using the *Interpolation Options* command in the *Interpolation* menu.
- 5. Interpolate the data set to the second mesh by selecting the *to Mesh* command from the *Interpolation* menu.

At this point, both data sets will be associated with the second mesh and the *Data Calculator* can be used to compute the difference between the two data sets. This same sequence of steps can be used to create initial conditions for a modified mesh to avoid repeated spin down (warm up) cycles for a network.

#### 4.2.2 Material -> Feature

The *Material* -> *Feature* command in the *Mesh Data* menu is used to create a feature coverage from an existing mesh. The material boundaries are converted to arcs and polygons created to represent each material region. The original nodal data should be shaved as a scatter point before regenerating new data from the coverage.

## 4.3 Mesh Display Options

The user controls which components of the mesh are displayed via the display options. The display options can be set by selecting the *Display Options* command in the *Display* Menu. Most of the items in the dialog are toggle boxes. If the toggle for a component of the mesh is set, the component is displayed when the mesh is re-drawn. The color, pattern, font, etc., used to display the component can be set using the popup display attribute editors. The appropriate editor for a component is invoked by clicking on the button to the left of the toggle box.

The mesh display options are as follows:

- The *Nodes* item is used to display mesh nodes. A small circle is drawn at each node.
- The *Elements* item is used to display elements of the mesh. The elements edges are drawn using the default color for elements. The *Materials* (see below) displayed also affects the appearance of the elements.
- The *Contours* item is used to display contours computed using the active scalar data set.
- The *Velocity Vectors* item is used to display a vector at each node using the active vector data set.
- The Nodestrings item is used to display nodestrings. Nodestrings are drawn
  in either the default nodestring color, or the color specified for a specific type
  of nodestring. The specific types of nodestrings are dependent on which
  model is being used, and therefore, these colors are set in the display options
  of the appropriate model menu.
- The *Mesh Quality* item is used to display potential problems with the current mesh. Elements that violate a quality criterion are highlighted in a color indicating the criterion which is violated by the element. By clicking on the options button to the left of this item the Mesh Quality Options dialog appears and allows the user to set the criteria.
- The *Background Grid* item is used to display a background grid for evenly distributing points or serving as a guide.
- The *Background Color* item is used to set the color of the background.
- The *Mesh Boundary* item is used to display a solid line around the perimeter of the mesh. Displaying the boundary is useful when contours are being displayed with the element edges turned off.

- The *Materials* option causes the elements to be drawn with the interior filled in with the color of the material associated with each element. This element color for the edges is drawn around the displayed material color.
- The *Material Boundary* item is used to display a solid line around the perimeter of zones of elements with a common material type.
- The *Material Numbers* item is used to display the ID associated with the material of each element.
- The *Node Numbers* item is used to display the ID associated with each node next to the node.
- The *Element Numbers* item is used to display the ID associated with each element at the centroid of the element.
- The *Nodal Elevations* item is used to display the z coordinate of each node next to the node.
- The *Erase Behind Labels* item is used to specify that the region behind the labels will be cleared to the background color to make the label more visible.
- The *Curved Mesh* item is used to specify that non-linear element edges be displayed using the quadratic basis functions associated with the element.

#### 4.4 Mesh Generation

The *Mesh Module* of *SMS* allows the user to perform two principal functions. First, the user creates and edits meshes, including boundary and global conditions. The second function, is to visualize the results of analysis. *SMS* provides tools to generate and edit the mesh via node and element operations. The creation of a finite element network requires the user to provide bathymetric information, and define the extremities of the network and points of interest inside the network. One set of data points may be used to define both the bathymetry and the network geometry, or they may be defined separately.

Digitized or surveyed points may be imported to provide bathymetry. If they are not intended for element creation, this data should be converted to scatter points (see section 4.2.1). If the data points also define the geometry of the network, the triangulate tool can be used to generate elements (see section 4.7.1), and then the mesh can be edited using the tools described in this chapter.

The Map Module provides tools for defining the geometry independently from the bathymetry (see chapter 6). The bathymetry can then be combined with the geometry using scatter point interpolation (see chapter 5). This process is also described in Lesson 2 of the tutorials.

### 4.5 Node Operations

Nodes are the basic building blocks of elements, and therefore for meshes. Nodes are also utilized in *SMS* as reference when constructing nodestrings, or assigning many types of boundary conditions.

### 4.5.1 Creating Nodes

New nodes are created by selecting the *Create Nodes* tool from the *Tool Palette* and clicking in the drawing area of the window where the new node is to be located. The default parameters governing the creation of new nodes can be selected using the *Node Options* dialog (see Figure 4.1), which is displayed by selecting the *Node Options* command from the *Nodes* menu. The top portion of the *Node Options* dialog includes three toggle boxes. The top toggle box entitled *Interpolate function from existing nodes* controls the z value for new nodes that are in the interior of the mesh. If this toggle is selected, any inserted new node is assigned a z value that is interpolated from the enclosing element. If the new node is not in the interior of an existing element, or this toggle is off, the radio group in the center of the dialog is used to specify whether to use a default z value for the new node or to have *SMS* prompt the user for the z value every time a new node is created.

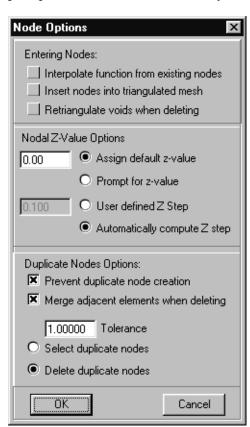


Figure 4.1 Node Options Dialog.

If the check box entitled *Insert nodes into triangulated mesh* is selected, any new node that lies in a region of the mesh elements will automatically be incorporated into the mesh. New nodes in quadrilateral elements result in those elements being split into triangular elements.

The *Retriangulate voids when deleting* toggle box controls what happens to the mesh when a node is deleted. If this toggle is set, the neighboring nodes are used to triangulate the void created by the node deletion. Otherwise, the void is left in the mesh.

In the center portion of the *Node Options* dialog two separate actions are controlled. The top two toggles correspond to the Z value assigned to a newly created node. The user may either enter a default Z value or ask to be prompted for the Z value at the creation of a new node. The bottom two toggles correspond to the step in Z value within the edit window (see section 2.5.1). The user can assign the value of the step in Z value or allow the computer to automatically compute a Z step. This step corresponds to the step made by clicking on the arrows in the edit box.

The bottom portion of the *Node Options* dialog deals with the handling of duplicate nodes. The *Tolerance* field allows the user to specify a distance that defines how close two nodes need to be to be considered duplicate. There are two means of eliminating duplicate node. The *Prevent duplicate node creation* toggle causes *SMS* to check whenever a node is created to ensure it is not a duplicate. This is the easiest method to prevent duplicate nodes, but is not entirely fail safe and can cause execution to slow down if a large mesh is in memory. The other means of making sure no duplicate nodes exist is to perform a duplicate node check. The bottom radio group allows the user to control whether duplicate nodes are selected or automatically deleted.

#### **Node Interpolation**

A set of new nodes can be interpolated between two previously selected nodes by selecting the *Interp Nodes* item from the *Nodes* menu. The locations and elevations of the new nodes are based on an interpolation of the coordinates of the two selected nodes. This interpolation is controlled by the *Node Interpolation Options* dialog (see Figure 4.2), which is also invoked from the *Nodes* menu.

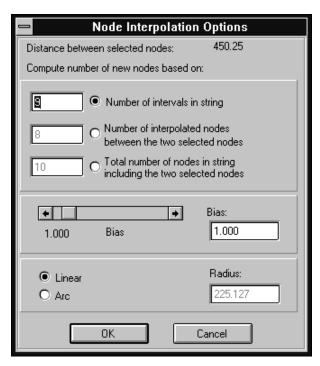


Figure 4.2 Node Interpolation Options.

The number of new nodes can be specified in three ways. The method used is controlled by the radio group in the middle of the *Node Interp Options* dialog. The user specifies either the number of intervals, the number of new nodes, or the number of nodes including the two end point nodes. The location of the nodes are controlled by the *bias* scroll bar and the *Linear/Arc* radio group at the bottom of the dialog. The bias controls how evenly the nodes are distributed. The value of the bias specifies the proportional distance between the last two nodes compared to the distance between the first two nodes. For example, if the bias value is 2.0, the distance between the last two nodes is twice the distance between the first two nodes. The first node is the node selected first. The *Linear/Arc* radio group controls whether the new nodes will be distributed along a straight line between the end points, or spread along an arc. If an arc is used, it will have a radius specified in the lower right corner of the dialog. The default radius is half the distance between the two points. The arc will proceed in a counter clockwise direction from the first node selected to the second node. An illustration showing how interpolated nodes are added is shown in Figure 4.3.

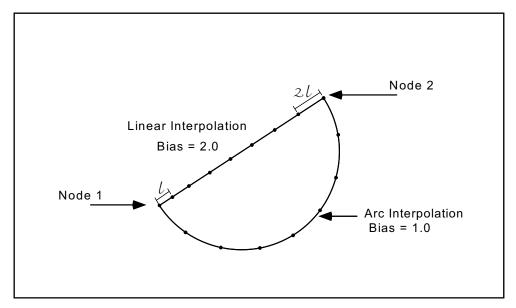


Figure 4.3 Interpolation Examples.

If a new node happens to fall in the interior of an element and the *Insert nodes into triangulated mesh* option in the *Node Options* dialog has been set, then the interpolated nodes are automatically incorporated into the mesh.

An additional use of the *Node Interp Options* dialog is to measure the distance between two nodes. This value is displayed in the dialog. If a user is only interested in finding the distance between the two selected nodes, he should select *Cancel* when exiting the dialog to avoid performing an interpolation.

### 4.5.2 Find Operations

SMS provides tools for the user to find specific nodes in several ways. These include:

- Find node from ID.
- Find node nearest to location.
- Find duplicate nodes (within tolerance).
- Find duplicate nodes at creation.

#### **Find Node**

Occasionally it is necessary to locate a node with a specific ID, or in a specific location. The desired node, corresponding to a user-specified ID or location can be located by selecting the  $Find\ Node$  item from the Nodes menu. The user is prompted for a node ID or an (x,y) coordinate and the node is selected. In the case of finding a node from location, the node nearest the specified location is found. Any previously selected nodes are unselected.

### **Find Duplicate Node**

Duplicate nodes can be found automatically by selecting the *Find Duplicates* command from the *Nodes* menu. The check uses the current node consolidation tolerance specified in the *Node Options* dialog (see Figure 4.1). Two nodes are considered to be duplicates if the xy distance between them is less than or equal to the specified tolerance. The user can also specify whether the duplicate nodes are to be automatically deleted or simply selected.

If the *Prevent Duplicate Node Creation* toggle has been enabled, this check will generally have no effect.

# 4.5.3 Editing Nodes

A mesh is never generated in final form. It always needs refinement, adjustments, and additional editing. *SMS* allows you to edit the nodes of a mesh interactively. The user can drag nodes by picking the *Select Node* tool from the *Tool Palette*, unlocking the nodes with the *Nodes Locked* item in the *Nodes* menu, and then clicking on and dragging the selected node.

## 4.5.4 Deleting Nodes

Nodes are deleted by selecting the nodes to be deleted using the *Select Node* tool from the *Tool Palette*, and hitting the *DELETE* key, *BACKSPACE* key, or selecting *Delete* from the *Edit* menu. If the *Retriangulate Voids When Deleting* flag is specified in the *Node Options* dialog (see Figure 4.1), and the node is included in the mesh, the resulting void is filled by triangulating the surrounding nodes.

# 4.5.5 Interpolating Nodal Boundary Conditions

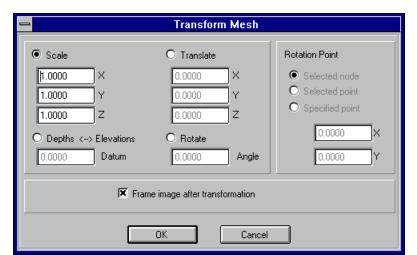
If two non-adjacent boundary nodes have boundary conditions of the same type, the user can assign nodal boundary conditions to all nodes between the two by selecting the *Interpolate Nodal BC* command from the *Node* menu. *SMS* creates a temporary nodestring connecting the two nodes. The nodestring goes counter clockwise around the boundary of the mesh, unless it is shorter to go clockwise, in which case *SMS* asks the user if that is the direction wanted.

# 4.6 Transforming Mesh

In some cases, it may be desirable to transform part or all of the mesh. The user can edit some or all of the geometry using the *Transform Mesh* dialog (see Figure 4.4). This is a useful tool if more than one region of the mesh have similar geometric characteristics. The user can construct one such region, and save it to a file. Then the

region can be duplicated by importing that file multiple times, and applying a different transformation to the data each time it is imported. First the user must selects the portion of the mesh (nodes) to be transformed. If the transformation is to apply to the entire mesh, no selection is necessary. Then the user selects the *Transform Mesh* item from the Nodes menu. This causes the *Transform Mesh* dialog to be displayed. The user then selects the type of transformation (scale, translation, rotation, or adjustment of the datum) and enters the appropriate parameters:

- Scale: User specifies scaling factors for the X, Y, and/or Z directions.
- Translate: User enters  $\Delta$  values for X, Y, and Z.
- Rotate: User defines a center of rotation. All other transformed points will be
  rotated about this center. The center can be defined as an (X,Y) point, a node
  or scatter point to rotate around. The user also specifies the angle (in
  degrees). All data points will be rotated the specified angle counter clockwise
  around the specified center of rotation.
- Depths <-> Elevations: *SMS* converts the nodal Z values from depths to elevations and vice versa. The user inputs a datum value to be used in the conversion process. This conversion is governed by the equation:



depth + elevation = datum

Figure 4.4 Transform Mesh Dialog.

# 4.7 Element Operations

Once nodes exist, the next step in mesh creation is the definition of elements using the nodes as corners. Elements can be created one at a time using the *Create Elements* tool. However, this process is labor intensive, so it is usually more convenient to use an automatic meshing technique to construct a finite element mesh.

Several meshing methods are provided in *SMS*. Triangulation and rectangular and triangular patches. The *Map Module* (chapter 6) provides fully automatic mesh generation. The *Mesh Module* provides partially automated and manual mesh generation. Techniques supported in the *Mesh Module* elements menu include:

## 4.7.1 Triangulation

New elements can be constructed by triangulating a set of nodes. The nodes are connected with a series of edges to form a triangulated mesh as shown in Figure 4.5. A set of nodes can be triangulated by selecting the *Triangulate* command from the *Elements* menu. The triangular elements this method creates will be linear if there are other linear elements already existing, or if a linear mesh has just been deleted. Otherwise, this method will create quadratic triangular elements. If no elements exist, and a specific type of element is desired, the user can first select either the three node triangle tool or the six node triangle tool and triangulate. If no nodes have been selected, all of the nodes are triangulated.

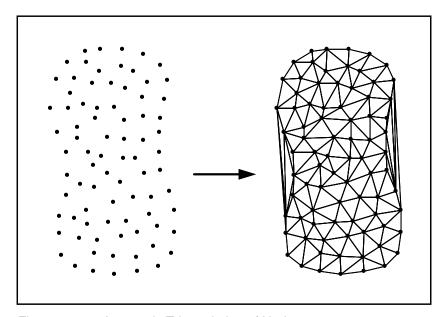


Figure 4.5 Automatic Triangulation of Nodes.

The algorithm that is used to triangulate nodes ensures that the triangulated mesh satisfies the Delauney criterion. The Delauney criterion is met, if the circumcircles of the triangles do not enclose any of the nodes of the mesh. The circumcircle of a triangle is the circle that passes through the three vertices of the triangle (see Figure 4.6).

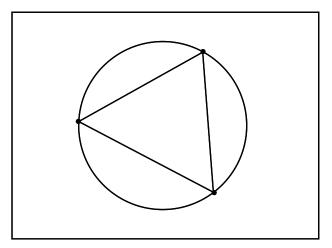


Figure 4.6 Circumcircle of a Triangle.

## 4.7.2 Element Options

Several operations involving elements require user specified parameters. These parameters are specified in the *Elements Options* dialog (see Figure 4.7). This dialog is divided into three sections.

### **General Options**

The *General Options* section which allows the user to specify parameters for general element operations.

- The *Display triangulation* toggle lets the user watch the triangulation process. By selecting this toggle, the process of creating the triangular elements is displayed. If this toggle is not selected, the display is updated at the completion of the triangulation. While the triangulation process is interesting to watch, the graphical display does slow down performance. This may be an issue with large meshes.
- The Select thin triangle aspect ratio allows the user to set a tolerance which defines what a thin triangle is (see page 4-21). The user specifies an aspect ratio (width/length). If a triangle's short dimension divided by its long dimension is less than this value, the triangle is considered to be thin.
- The *Merge triangle feature angle* controls the automatic merging of triangles into quadrilaterals (see page 4-22). The *Preserve material boundaries* toggle influences triangle merging along material boundaries. If this toggle is set, two triangles which reference different materials cannot be merged.
- The Smooth nodestring feature angle data entry is discussed on page 4-25.

#### **Relax Elements**

The *Relax Elements* section of the *Elements Options* dialog allows the user to specify parameters for the mesh relaxation operation (page 4-24).

- The *Number of iterations* data entry allows the user to specify a number of iterations to perform during relaxation.
- The *Interpolate Z from existing mesh* toggle controls the way the z values are modified. If this toggle is selected the nodes z values conform to the existing elements. If the toggle is not selected, the z values of each node stays the same through the relax process.
- The *Preserve material boundaries when relaxing* toggle restricts the relaxation process to nodes that do not lie on material boundaries.

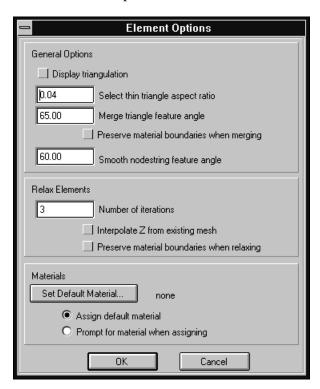


Figure 4.7 Element Options Dialog.

#### **Materials**

The *Materials* section of the *Elements Options* dialog allows the user to control how materials are assigned to elements. The user can choose to have the default material assigned to newly created elements or have *SMS* ask the user each time an element is created. The specified material is called the default material. The current default material is shown in this dialog. Assigning the default material is described in Section 2.8.4.

### 4.7.3 Rectangular Patches

Rectangular and triangular patches of elements can be created by creating the nodes on the boundary of a region and filling the interior of the patch automatically. This may be the case when a user defines a mesh by first defining the boundaries between different types of bed material. The regions defined by these boundaries must be filled with elements. The use of patches is especially applicable when the data points gathered are along lines such as cross sections. Because a patch infers the shape of the interior elements from its boundaries, it tends to interpolate from one boundary to the next. This may reduce the amount of data points required to generate a good mesh. A sample rectangular patch is shown in the *Rectangular Patch* dialog in Figure 4.8.

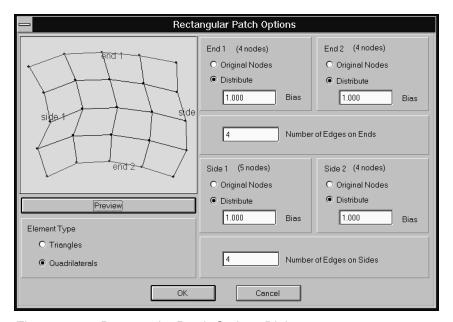


Figure 4.8 Rectangular Patch Options Dialog.

The coordinates of the new nodes on the interior of the patch are computed by constructing a partially bicubically blended Coon's patch based on the curves defined by the four edges of the patch. This ensures that the location and elevation of the interior nodes are smoothly interpolated from the nodes making up the perimeter of the patch.

A rectangular patch of elements can be constructed using the following steps:

- Create a nodestring along each of the four edges of the boundary of the patch
  to be constructed. The nodestrings can be created in any direction. Midside
  nodes (nodes at the midpoint of the edges of quadratic elements) can be
  selected or ignored. The only restriction is that the four nodestrings must
  close to form a rectangular region.
- Choose the Select Nodestring tool, and select the four nodestrings. The order
  of selection is not important.

- Select the *Rectangular Patch* command from the *Elements* menu.
- *SMS* will display the *Rectangular Patch* dialog, prompting the user to select the type of element used to fill the patch (triangular or quadrilateral). The user is also prompted to define the node distribution along the edges of the patch. Nodes can be distributed along the edges as long as no elements exist on that edge, and the number of nodes is consistent with the opposing edge.
- At this point the user can preview the effects of the current parameters by clicking on the *Preview* button. *SMS* computes the new elements, and displays them in the preview window.

As the patch is being constructed, the elements are checked to see if they are ill-formed and to make sure they do not overlap surrounding elements. The new elements are guaranteed to conform properly (linear on linear, quadratic on quadratic) to the elements adjacent to the patch. If any problems are detected, a message is given and the patch creation is aborted.

If the rectangular region specified for the patch is highly irregular in shape, the patch creation process may fail and abort. In such cases, the region can typically be meshed by subdividing the patch into a number of smaller patches (by adding extra rows of nodes) and filling in each of the smaller patches with elements.

# 4.7.4 Triangular Patches

Triangular patches of elements can be created to fill voids with three boundaries. A sample triangular patch in the *Triangular Patch Options* dialog is shown in

Figure 4.9. The coordinates of the new nodes on the interior of the patch are computed by blending the coordinates of the three edges of the patch.

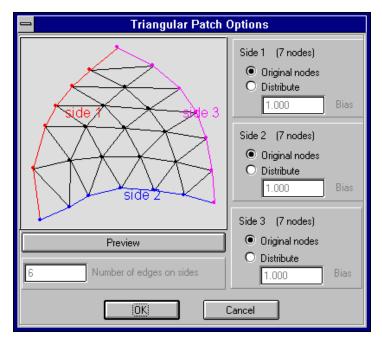


Figure 4.9 Triangular Patch Options Dialog with Patch.

The following steps can be taken to create a triangular patch of elements:

- Create a nodestring along each of the three edges of the boundary of the
  patch to be constructed. The nodestrings can be created in any direction.
  Midside nodes (nodes at the midpoint of the edges of quadratic elements) can
  be selected but are ignored. The only restriction is that the three nodestrings
  must close to form a triangular region.
- Select the *Select Nodestring* tool, and select the three nodestrings. The order of selection is not important.
- Select the *Triangular Patch* command from the *Elements* menu.
- *SMS* will display the *Triangular Patch Options* dialog, prompting the user to define the distribution of the nodes along the three edges. Nodes may be distributed if no elements exist on the edge, and the number of nodes is consistent with the other edges.
- At this point the user can preview the effects of the current parameters by clicking on the *Preview* button. *SMS* computes the new elements, and displays them in the preview window.

If the triangular region specified for the patch is highly irregular in shape, the patch creation process may fail and abort. In such cases, the region can typically be meshed by subdividing the patch into a number of smaller patches (by adding extra rows of nodes) and filling in each of the smaller patches with elements.

### 4.7.5 Find Operations

As with nodes, it may become necessary to locate specific elements *SMS* provides three means of finding specific elements. These include:

- 1. Selecting thin triangles on the mesh boundary.
- 2. Finding an element by ID.
- 3. Finding an element by location.

### **Select Thin Triangles**

During the process of triangulation, a mesh of triangular elements is created around the existing nodes. The boundaries of this mesh is defined by the convex hull of these nodes. This generally is not the desired boundary for the problem mesh. As a result, the user must first insure that the desired mesh boundary follows element edges by swapping element edges (see page 4-5) or inserting breaklines (see page 4-22). Then the user must select the elements outside the desired mesh boundary and delete them. If three nodes on the mesh boundary are very close to being collinear, the convex hull resulting from triangulation can create very long skinny triangles on the outside of the desired boundary. These triangles may be so thin that when looking at the mesh, they appear invisible until the user zooms in on them several times. Since they can exist at any point along the mesh boundary, checking the entire boundary for such elements can be tedious. However, due to the nature of skinny triangles in finite element analysis, they can often lead to instabilities in the mesh. In addition, since they are on the boundary, they can lead to confusion when creating boundary conditions. For these reasons, these triangles must be deleted. SMS simplifies this process by providing a method for automatically selecting such triangles using the Select Thin Triangles item from the Elements menu. The triangles are selected in the following manner: The triangles on the outer boundary are checked first. If the aspect ratio of the triangle is less than a critical value, the triangle is selected and the triangles adjacent to the triangle are then checked. The process continues inward until none of the adjacent triangles violate the minimum aspect ratio.

Thin triangles in the interior of the mesh will not be selected. The user must clean up these poor elements by hand since deletion of interior triangles would result in gaps in the mesh. To locate skinny triangles in the interior of the mesh the user should use the mesh quality display option to highlight all thin triangles. This option causes all triangles which meet the aspect ratio criteria to be highlighted regardless of their location in the mesh.

#### **Find Element**

Occasionally it is necessary to locate an element with a specific ID. The element corresponding to a user-specified ID can be located by selecting the *Find Element* item from the *Elements* menu. The user is prompted for an element ID and the element is selected. Any previously selected elements are unselected.

Alternatively, the user may select a coordinate location and *SMS* will draw the element closest to that location in red.

#### 4.7.6 Breaklines

A breakline is a feature line or polyline representing a ridge or some other feature that the user wishes to preserve in a mesh. In other words, a breakline is a series of edges that the triangles should conform to or not intersect (see Figure 4.10).

Breaklines can be processed using the *Add Breaklines* command from the *Elements* menu. Before selecting this command, one or more nodestrings which define sequences of nodes defining the breakline(s) should be selected using the *Select Nodestrings* tool from the *Tool Palette*.

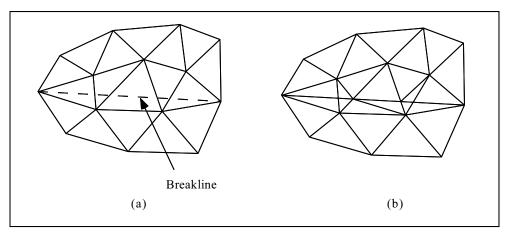


Figure 4.10 (a) Mesh and Breakline. (b) Mesh After Processing Breakline

As each breakline is processed, the triangles intersected by the breakline are modified by adding new nodes at necessary locations to ensure that the edges of the triangles will conform to the breakline. The elevations of the new nodes are based on a linear interpolation of the breakline segments. The locations of the new nodes are determined in such a way that the Delauney criterion is satisfied.

# 4.7.7 Merge Triangles

The triangulation operation results in a mesh composed entirely of triangles. In some cases it is desirable to have the mesh composed primarily of quadrilateral elements. Quadrilateral elements result in a more concise mesh which leads to faster solutions, and quadrilateral elements are often more numerically stable. To address this need, two options are provided for converting triangular elements to quadrilateral elements.

The *Merge Triangles* command in the *Elements* menu can be used to automatically merge pairs of adjacent triangular elements into quadrilateral elements. The *Merge Triangles* command uses the minimum interior angle specified in *Elements Option* dialog (see Figure 4.7). This angle should be between 0 and 90 degrees. If no

elements are selected, all of the triangular elements in the mesh are then processed. If some elements have been selected, only the selected elements are processed as follows:

- The set of elements to be processed is traversed one element at a time. Each triangular element that is found is compared with each of its three adjacent elements. If the adjacent element is a triangle, the trapezoid formed by the triangle and the adjacent triangle is checked.
- Each of the four interior angles of the trapezoid is computed and compared to the specified minimum interior angle. If all of the angles are greater than the user-specified minimum interior angle, then the two triangles are merged into a single quadrilateral element.

This process is repeated for all of the elements. *SMS* actually does a series of merges when this command is issued. First a merge is performed with a large minimum interior angle (85°). The merging is then repeated with progressively smaller minimum interior angles (80°, 75°, 70°, 65°, etc.) until the user specified angle is reached. This ensures that a triangle is not merged with one adjacent triangle when a merge with another of its other adjacent triangles would have resulted in a quadrilateral element with a better aspect ratio.

The merging scheme will not always result in a mesh composed entirely of quadrilateral elements. Some triangular elements are often necessary in highly irregular meshes to provide transitions from one region to the next.

The other option for merging triangles involves the use of the *Merge/Split* tool in the *Tool Palette*. If the *Merge/Split* tool is selected, clicking on a triangle edge with the mouse cursor will cause the two triangular elements adjacent to the edge to be merged into a quadrilateral element provided that the trapezoid formed by the two triangles is not concave.

This tool can be used to manually merge triangles, one pair at a time, rather than using the automatic scheme described above.

The manual method is also useful to edit or override the results of the automatic merging scheme in selected areas. The *Merge/Split* tool can also be used to undo a merge. A quadrilateral element can be split into two triangles by clicking anywhere in the interior of the element. This tool is useful if a pair of triangles are inadvertently merged.

### 4.7.8 Split Quadrilaterals

Occasionally it is necessary to split quadrilateral elements into triangular elements.

Two options are provided for splitting quadrilateral elements:

- The *Split Quads* command in the *Elements* menu can be used to split a group of quadrilateral elements into triangular elements. If no elements are selected, all of the quadrilateral elements in the mesh will be split. If some elements have been selected, only the selected quadrilateral elements will be split.
- The other option for splitting quadrilateral elements involves the use of the *Merge/Split* tool in the *Tool Palette*. If the *Merge/Split* tool is selected, clicking anywhere in the interior of a quadrilateral element with the mouse cursor will cause the element to be split into two triangles. The shortest diagonal through the quadrilateral is chosen as the common edge of the two new triangular elements.

#### 4.7.9 Element Conversion

#### Linear <-> Quadratic

Linear elements (three node triangles and four node quadrilaterals) can be converted to quadratic elements (six node triangles and eight node quadrilaterals) and vice versa by selecting the *Linear*<->*Quadratic* item from the *Elements* menu.

#### Quad8 <-> Quad9

Some numerical models, such as *FESWMS*, support quadratic quadrilateral elements with center nodes (i.e. nine node elements), while others, such as the version of *RMA2* distributed through WES, do not. Therefore, it is useful in some cases to convert between the two. This is accomplished by selecting the *Quad8*<->*Quad9* item from the *Elements* menu.

# 4.7.10 Refining Elements

In some cases, a mesh does not have enough elements in a certain area to insure stability. Rather than inserting supplemental nodes and re-creating the mesh, it is possible to refine a selected region of the mesh using the *Refine Elements* command in the *Elements* menu. This quadruples the mesh density of a selected area of the mesh. If no elements are selected, the entire mesh is refined. The elevation of the new nodes is interpolated from the existing nodes. Transitions from areas of low mesh density to areas of high mesh density are made using triangular elements. When a mesh is refined the numbering is changed, therefore a mesh should be renumbered (section 4.7.13) after elements have been refined.

# 4.7.11 Relaxing Elements

After refining part of a mesh, the transition between the new area and the old area may violate the 50% size difference rule for adjoining elements, as explained in Chapter 2 of the TABS Primer. In such cases, the *Relax Elements* command can

improve the mesh by iteratively moving the nodes of a selected set of elements to the centroid of the elements adjacent to each node. The user must select the elements to be affected, or *SMS* will refine the entire mesh. The user can specify options to preserve the existing bathymetry, and the existing material boundaries in the *Element Options* dialog. The natural result of relaxing elements is that quadrilateral elements are shaped closer to squares and triangles are shaped closer to equilateral triangles.

## 4.7.12 Smooth Edges

When a quadratic element is created, the midside nodes are located at the midpoint of the straight line between the corner nodes. This is generally okay, but the angular corners resulting from such elements can cause a numerical problem along the boundaries. This is because the abrupt change in boundary direction can cause the analysis to try to make an abrupt change in flow direction (see Figure 4.11a). Such a discontinuity may result in inaccuracy in the numerical model sometimes referred to as a mass loss. This can be thought of as water flowing out of the mesh. On the positive side, this can be detected by the numerical model to give the user a warning that it is happening. To correct the problem, the abrupt change in flow direction needs to be minimized. One means of doing this is to refine the elements, replacing one abrupt change with several smaller changes. This also increases the complexity of the model. Another option is to curve the edges of the elements. Moving the midside node of an element defines a curved edge. This curved edge creates a smoother boundary or transition from one element to the next and is therefore referred to as smoothing the boundary. An example is shown in Figure 4.11b. The discontinuity is reduced and the mesh becomes more stable. Normally, this is only a concern along the boundary of the mesh. However, if the analysis involves regions that dry out during the analysis, it is wise to smooth edges along the elements that may become the boundary of the wet portion of the mesh.

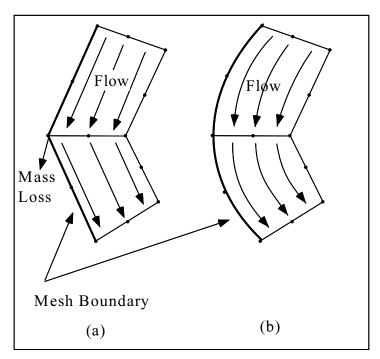


Figure 4.11 Example of Mesh Boundary (a) Before Smoothing and (b) After Smoothing

Midside nodes can be moved by dragging the nodes with the *Select Nodes* tool. As a node is dragged, the adjacent element edges are drawn using a quadratic isoparametric curve. In addition to manual editing, *SMS* has a simple algorithm that attempts to position midside nodes so that the iso-parametric curves have continuous slopes at the corner nodes.

The following steps can be taken to smooth a segment of node edges:

- Using the *Create Nodestring* tool, create the string of element edges to be smoothed.
- Using the *Select Nodestring* tool, select all nodestrings to be smoothed.
- Select the *Smooth Segment* command from the *Elements* menu.
- *SMS* attempts to smooth all edges with an angle of less than the *Smooth Edge Feature Angle* (specified in the *Element Options* dialog). Some edges cannot be automatically smoothed and must be done manually.

#### 4.7.13 Renumbering

The nodes and the elements can be efficiently renumbered by selecting the *Renumber* item from the *Elements* menu. A nodestring must be selected before renumbering the mesh (see the description of the *Select Nodestrings* tool above).

The selected nodestring is used to specify where the renumbering process begins. This nodestring is referred to as a GO string in TAB-MD and as a RESE string in *FESWMS*. All nodestrings not assigned as a boundary condition or flux string are preserved as a renumbering string. The "row" of elements and nodes adjacent to the selected string are numbered first. The elements and nodes adjacent to the first set of nodes and elements are numbered next, and so on until all of the nodes and elements have been renumbered.

The nodes and elements are renumbered in a sequence that can be envisioned as a "moving front" that passes through the mesh. Since the front proceeds from one set of elements to an adjacent set of elements, disjoint portions of the mesh will not be visited in the renumbering process. Unvisited nodes and elements are numbered arbitrarily.

After invoking the *Renumber* command, the user will be asked whether a band width or front width renumber is desired. The band width renumber attempts to count the number of equations related to each node on the front, and sweep the front in such a way as to minimize this number. The front width only considers the number of nodes along the front.

# 4.8 Assign Materials

Each element in the mesh has an associated material type. The default material ID can be set by bringing up the *Element Options* dialog in the *Elements* menu. Clicking the *Select Default Material* button allows the user to select the default material from one of the currently defined materials.

A new material can be assigned to an element or a set of elements by selecting the element(s) and then selecting the *Assign Material Types* command from the *Elements* menu. The element will be assigned the default material unless the *Prompt for material when assigning* button is chosen. This button is found in the *Element Options* dialog in the *Elements* menu. Deleting unused materials, creating new materials, and changing the material attributes is described in Section 2.8.4.

### 4.9 Model Menus

In the current version of SMS, 2D finite element analysis engine interfaces are available for *RMA2/RMA4*, *SED2D-WES*, HIVEL2D, RMA10 and *FESWMS*. Anything specific to a numerical model is accessed through the appropriate menu. These model specific attributes include the material attributes, boundary conditions, and model run control. For details on a specific model, refer to the appropriate chapter in this manual.

CHAPTER 5

# Scatter Point Module

The *Scatter Point Module* is used to interpolate from groups of scattered data points to the other data types (i.e., meshes, grids). Several interpolation schemes are supported including natural neighbor and inverse distance weighted.

Interpolation is useful for setting up input data for analysis codes. Generally, the data gathered from a site to be modeled will not be dense enough to create a quality mesh. Or it will be too dense to be processed in a reasonable amount of time. Interpolation allows the gathered data points to be used as background information. The user may then generate a base mesh in the Map Module (see Chapter 6). This base mesh lacks the bathymetry data to accurately represent the area being modeled. By interpolating from the scattered data to the mesh, the bathymetry can be assigned to the mesh.

Interpolation is also useful for comparison of different meshes of the same region. For example, a user may have two meshes representing a river reach. One without some feature such as bridge abutments or piers, and one with. Using interpolation, the data from one mesh can be converted to a scattered data set and then interpolated to the other mesh. Then the two data sets could be compared directly.

### 5.1 Scatter Point Sets

Points from which values are interpolated from are called scatter points. A group of scatter points is called a scatter point set. Each of the scatter points is defined by a set of xy coordinates.

Each scatter point set has a list of scalar data sets (vector data sets are not currently supported for scatter point groups). Each data set represents a set of values which can

be interpolated to a mesh or grid. When an interpolation command is selected, the active data set for the scatter point set is used in the interpolation process.

Multiple scatter point sets can exist in memory. One of the scatter point sets is always designated as the "active" scatter point set. Interpolation is performed from the active scatter point set only. The active scatter point set can be changed using the *Select Scatter Point Set* tool described below. Whenever a new scatter point set is read from a file or created, it becomes the active set.

# 5.2 Inputting Sets

Scatter point sets can be created by converting from other data types (i.e., meshes, grids). For example, if a finite element mesh is converted to a scatter point set (see Section 4.2.1), each of the nodes in the mesh become a scatter point and each of the scalar data sets associated with the mesh are copied to the data set list for the new scatter point set.

Scatter point sets can also be input from a text file. The file formats for scatter point sets are described in Section 14.3. Two types of file formats can be used to import scatter point sets: the XY format and the XYD format. With the XY format, only the xy coordinates for each point are contained in the file. Data values associated with each point must then be input through the *Data Browser* using a data set file. With the XYD format, the xy coordinates and one or more scalar values may be defined for each scatter point. When this type of file is imported, a data set is automatically created for each set of scalar values.

# 5.3 Saving Sets

Scatter point sets may be saved to a text file using the *Save* command from the *File* Menu. When a scatter point file is saved, the user must specify in the *Save* dialog whether the XY or XYD format will be used. If the XY format is chosen, only the xy coordinates of the scatter points are saved to the file. Any data sets associated with the scatter points must be saved using the *Export* option in the *Data Browser*. If the XYD format is chosen, the xy coordinates and the data sets are saved to a single file. However, since the XYD format is only used for steady-state data, only the current time step of each data set is saved. To save a complete dynamic data set, the XY format must be used in conjunction with the *Export* option in the *Data Browser* 

### 5.4 Tool Palette

The following tools are active in the dynamic portion of the *Tool Palette* whenever the *Scatter Point Module* is active.

# 5.4.1 Select Scatter Point

The *Select Scatter Point* tool is used to select individual scatter points for editing using the *Edit Window*.

# 5.4.2 Select Scatter Point Set

The Select Scatter Point Set tool is used to select entire scatter point sets for deletion or to designate the active scatter point set. When this tool is active, an icon appears at the centroid of the set for each of the scatter point sets. A scatter point set is selected by selecting the icon for the set.

A selected scatter point set can be made the active set by double clicking on the icon for the set or by selecting the *Make Active* command from the *Interpolate* menu.

# 5.5 Display Options

A scatter point set is displayed by drawing a symbol for each of the scatter points. The display options control the appearance of the symbol and the labels. Each scatter point set has its own display options. The display options can be set by selecting the *Display Opts* command from the *Display Menu*.

The scatter point display options are as follows:

- The *Scatter Point Numbers* item is used to display the scatter point ID number next to each scatter point.
- The *Scatter Point Symbols* item is used to display a symbol at the location of each scatter point. The button to the left of the item is used to bring up a dialog listing the available symbols. The color of each of the scatter points in a set may be changed from this dialog also.

The *Scatter Point* display options apply to the active scatter point set only. To change the symbols for a scatter point set other than the active set, the set must first be made active by double clicking on the set with the *Select Scatter Point Set* tool, or by selecting the set and selecting the *Make Active* command from the *Interpolation* menu prior to bringing up the *Display Options* dialog.

### 5.6 Scatter Point Conversion

Scatter point sets can be used to create other types of objects. When the new object is created, all data sets associated with the scatter point set are copied to the new object.

### 5.6.1 Scatter Points -> Mesh Nodes

The *Scatter Points -> Mesh Nodes* command creates a set of finite element nodes. These nodes can be triangulated by selecting the *Triangulate* command from the *Mesh Module*.

# 5.7 Interpolation

### 5.7.1 Interpolation From Scatter Point Sets

The data associated with the active time step of the active data set of the active scatter point set can be interpolated to another object. Before this can be done, the user must select an interpolation scheme and set all of the parameters for the selected scheme if other than default values are desired. During the interpolation process, a new data set is constructed for the target object containing the interpolated values.

The interpolation command to interpolate to a mesh is found in the *Interpolation* menu. The *to Mesh* command interpolates to the nodes of the finite element mesh.

## 5.7.2 Interpolation Options

Scatter point sets are used for interpolation to other data types such as grids, and meshes. Since no interpolation scheme is superior in all cases, several interpolation techniques are provided by *SMS*.

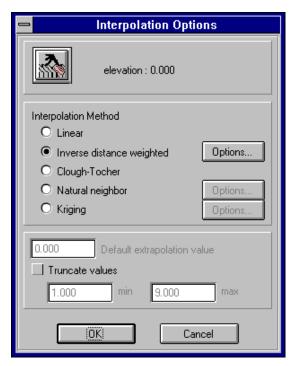


Figure 5.1 The Interpolation Options Dialog.

The interpolation option is selected using the *Interpolation Options* dialog (see Figure 5.1) which is accessed through the *Interp. Options* command in the *Interpolation* menu. Once an option is selected, that option is used for all subsequent interpolation commands.

Interpolation is always performed using the active scatter point set. Only the active data set and time step are interpolated. The active data set and time step can be selected using the *Data Browser* command in the *Data* menu or using the button at the top of the *Interpolation Options* dialog.

When interpolating a set of values, it is sometimes useful to limit the interpolated values to lie between a minimum and maximum value. For example, when interpolating contaminant concentrations, a negative value of concentration is meaningless. However, many interpolation schemes will produce negative values even if all of the scatter points have positive data values. This occurs in areas where the trend in the data is toward a zero value. The interpolation may extend the trend beyond a zero value into the negative range. In such cases it is useful to limit the minimum interpolated value to zero. Interpolated values can be limited to a given range by selecting the *Truncate values* option in the *Interpolation Options* dialog and entering a minimum and maximum interpolation value.

The available interpolation methods are listed in the *Interpolation Options* dialog. To the right of most of the method names is a button used to bring up a dialog for entering additional interpolation parameters specific to the selected interpolation method. The methods supported for interpolation are *linear*, *inverse distance* weighted, Clough - Tocher, and natural neighbor.

## 5.7.3 Linear Interpolation

If the *linear* interpolation scheme is selected, the scatter points are first triangulated to form a temporary triangular network. If the surface is assumed to vary linearly across each triangle, the network describes a piecewise linear surface which interpolates the scatter points. The equation of the plane defined by the three vertices of a triangle is as follows:

$$Ax + By + Cz + D = 0$$
....(5.1)

where A, B, C, and D are computed from the coordinates of the three vertices  $(x_1,y_1,z_1), (x_2,y_2,z_2), \& (x_3,y_3,z_3)$ :

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)....(5.2)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \dots (5.3)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)....(5.4)$$

$$D = -Ax_1 - By_1 - Cz_1$$
....(5.5)

The plane equation can also be written as:

$$z = f(x,y) = -\frac{A}{C} x - \frac{B}{C} y - \frac{D}{C}$$
 (5.6)

which is the form of the plane equation used to compute the elevation at any point on the triangle.

Since a triangular network only covers the convex hull of a scatter point set, extrapolation beyond the convex hull is not possible with the linear interpolation scheme. Any points outside the convex hull of the scatter point set are assigned the *Default extrapolation value* specified in the *Interpolation Options* dialog.

# 5.7.4 Inverse Distance Weighted Interpolation

One of the most commonly used techniques for interpolation of scatter points is *inverse distance weighted (IDW)* interpolation. Inverse distance weighted methods are based on the assumption that the interpolating surface should be influenced most by the nearby points and less by the more distant points. The interpolating surface is a weighted average of the scatter points and the weight assigned to each scatter point diminishes as the distance from the interpolation point to the scatter point increases.

Several options are available for inverse distance weighted interpolation. The options are specified using the *IDW Interpolation Options* dialog (see Figure 5.2).

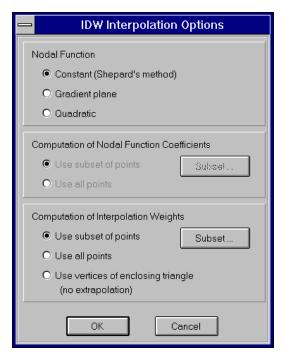


Figure 5.2 The IDW Interpolation Options Dialog.

### **Shepard's Method**

The simplest form of inverse distance weighted interpolation is sometimes called "Shepard's method" (Shepard 1968). The equation used is as follows:

$$F(x,y) = \sum_{i=1}^{n} w_i f_i .....(5.7)$$

where n is the number of scatter points in the set,  $f_i$  are the prescribed function values at the scatter points (e.g. the data set values), and  $w_i$  are the weight functions assigned to each scatter point. The classical form of the weight function is:

$$w_{i} = \frac{h_{i}^{-p}}{\sum_{j=1}^{n} h_{j}^{-p}}$$
 (5.8)

where p is an arbitrary positive real number called the power parameter (typically, p=2) and  $h_i$  is the distance from the scatter point to the interpolation point or

$$h_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$$
 .....(5.9)

where (x,y) are the coordinates of the interpolation point and  $(x_i,y_i)$  are the coordinates of each scatter point. The weight function varies from a value of unity at

the scatter point to a value approaching zero as the distance from the scatter point increases. The weight functions are normalized so that the weights sum to unity.

The effect of the weight function is that the surface will interpolate each scatter point and be influenced most strongly between scatter points by the points closest to the point being interpolated.

Although equation 5.8 is typically used for the weight function in inverse distance weighted interpolation, the following equation is used in *SMS*:

$$w_{i} = \frac{\left[\frac{R-h_{i}}{Rh_{i}}\right]^{2}}{n} \qquad (5.10)$$

$$\sum_{j=1}^{\infty} \left[\frac{R-h_{j}}{Rh_{j}}\right]^{2}$$

where h<sub>i</sub> is the distance from the interpolation point to scatter point i, R is the distance from the interpolation point to the most distant scatter point, and n is the total number of scatter points. This equation has been found to give superior results to equation 5.8 (Franke & Nielson, 1980).

The weight function is a function of Euclidean distance and is radially symmetric about each scatter point. As a result, when interpolating elevation, the interpolating surface is somewhat symmetric about each point and tends toward the mean elevation of the scatter points between the scatter points. Shepard's method has been used extensively because it is very simple.

#### **Gradient Planes Nodal Functions**

A limitation of Shepard's method is that the interpolating surface is a simple weighted average of the data values of the scatter points and is constrained to lie between the extreme values in the data set. In other words, the surface will not infer local maxima or minima implicit in the data set. This problem can be overcome by generalizing the basic form of the equation for Shepard's method in the following manner:

$$F(x,y) = \sum_{i=1}^{n} w_i Q_i(x,y) .... (5.11)$$

where  $Q_i$  are nodal functions or individual functions defined at each scatter point (Franke 1982; Watson & Philip 1985). The value of an interpolation point is calculated as the weighted average of the values of the nodal functions at that point. The standard form of Shepard's method can be thought of as a special case where horizontal planes (constants) are used for the nodal functions. The nodal functions can be sloping planes that pass through the scatter point. The equation for the plane is as follows:

$$Q_{i}(x,y) = f_{x}(x-x_{i}) + f_{y}(y-y_{i}) + f_{i}....(5.12)$$

where  $f_x$  and  $f_y$  are partial derivatives at the scatter point that have been previously estimated based on the geometry of the surrounding scatter points. Gradients are estimated in SMS by first triangulating the scatter points and computing the gradient at each scatter point as the average of the gradients of each of the triangles attached to the scatter point.

The planes represented by equation 5.12 are sometimes called "gradient planes." By averaging planes rather than constant values at each scatter point, the resulting surface infers extremities and is asymptotic to the gradient plane at the scatter point rather than forming a flat plateau at the scatter point.

#### **Quadratic Nodal Functions**

The nodal functions used in inverse distance weighted interpolation can also be higher degree polynomial functions constrained to pass through the scatter point and approximate the nearby points in a least squares manner. Quadratic polynomials have been found to work very well (Franke & Nielson 1980; Franke 1982). The resulting surface reproduces local variations implicit in the data set, is very smooth, and will approximate the quadratic nodal functions near the scatter points. The equation used for the quadratic nodal function centered at point k is as follows:

$$Q_k(x,y) = a_{k1} + a_{k2}(x - x_k) + a_{k3}(y - y_k) + a_{k4}(x - x_k)^2$$

$$+ a_{k5}(x - x_k)(y - y_k) + a_{k6}(y - y_k)^2 \dots (5.13)$$

To define the function, the six coefficients  $a_{k1}...a_{k6}$  must be found. Since the function is centered at the point k and passes through point k, we know beforehand that  $a_{k1}=f_k$  where  $f_k$  is the function value or z-value at point k. The equation simplifies to:

$$Q_{k}(x,y) = f_{k} + a_{k2}(x - x_{k}) + a_{k3}(y - y_{k}) + a_{k4}(x - x_{k})^{2}$$

$$+ a_{k5}(x - x_{k})(y - y_{k}) + a_{k6}(y - y_{k})^{2} .... (5.14)$$

Now there are only five unknown coefficients. The coefficients are found by fitting the quadratic to the nearest  $N_Q$  scatter points in a weighted least squares fashion. In order for the matrix equation used to solve for the coefficients to be stable, there should be at least five scatter points in the set.

#### Interpolation Subsets

In the *IDW Interpolation Options* dialog shown in Figure 5.2, an option is available for using a subset of the scatter points (as opposed to all of the available scatter points) in the computation of the nodal function coefficients and in the computation of the interpolation weights. Using a subset of the scatter points drops distant points from consideration since they are unlikely to have a large influence on the nodal

function or on the interpolation weights. Using a subset can speed up the computations since less points are involved.

If the *Use subset of points* option is chosen, the *Subsets* button can be used to bring up the *Subset Definition* dialog shown in Figure 5.3. Two options are available for defining which points are included in the subset. In one case, only the nearest N points are used. In the other case, only the nearest N points in each quadrant are used (see Figure 5.4). This approach may give better results if the scatter points tend to be clustered.

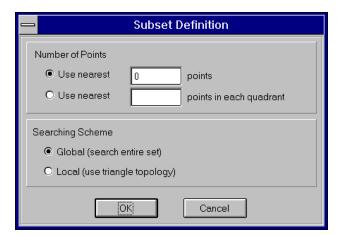


Figure 5.3 The Subset Definition Dialog.

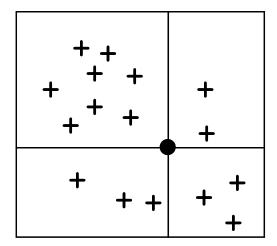


Figure 5.4 The Four Quadrants Surrounding an Interpolation Point.

If a subset of the scatter point set is being used for interpolation, a scheme must be used to find the nearest N points. Two methods for finding a subset are provided in the *Subset Definition* dialog: the global method and the local method. With the global method, each of the scatter points in the set are searched for each interpolation point to determine which N points are nearest the interpolation point. This technique is fast for small scatter point sets but may be slow for large sets.

With the local scheme, the scatter points are triangulated to form a temporary triangular network before the interpolation process begins. To compute the nearest N points, the triangle containing the interpolation point is found and the triangle topology is then used to sweep out from the interpolation point in a systematic fashion until the N nearest points are found. The local scheme is typically much faster than the global scheme for large scatter point sets.

### **Local Weighting Method**

As mentioned above, it is possible to localize the search for the nearest N scatter points to the interpolation point using the topology of a triangular network constructed from the scatter points. Yet another scheme is available for making the interpolation process a local scheme by taking advantage of network topology (Franke & Nielson, 1980). With this technique, the subset of points used for interpolation consists of the three vertices of the triangle containing the interpolation point. The weight function or blending function assigned to each scatter point is a cubic S-shaped function (see Figure 5.5a). The fact that the slope of the weight function tends to unity at its limits ensures that the slope of the interpolating surface will be continuous across triangle boundaries.

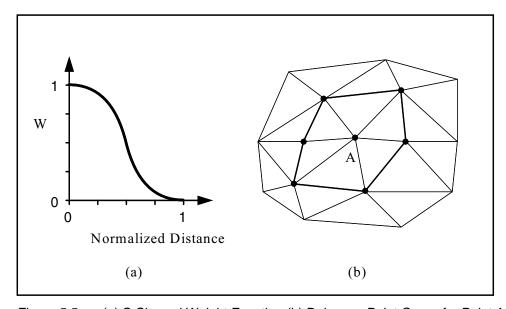


Figure 5.5 (a) S-Shaped Weight Function (b) Delauney Point Group for Point A.

The influence of the weight function extends over the limits of the Delauney point group of the scatter point. The Delauney point group is the "natural neighbors" of the scatter point, and the perimeter of the group is made up of the outer edges of the triangles that are connected to the scatter point, as shown in Figure 5.5b. The weight function varies from a weight of unity at the scatter point to zero at the perimeter of the group. For every interpolation point in the interior of a triangle, there are three nonzero weight functions (the weight functions of the three vertices of the triangle). For a triangle T with vertices i, j, & k, the weights for each vertex are determined as follows:

$$\begin{split} W_{i}(x,y) &= b_{i}^{2}(3-2b_{i}) \\ &+ 3 \frac{b_{i}^{2}b_{j}b_{k}}{b_{i}b_{j}+b_{i}b_{k}+b_{j}b_{k}} \left\{ b_{j} \left[ \frac{||e_{i}||^{2}+||e_{k}||^{2}-||e_{j}||^{2}}{||e_{k}||^{2}} \right] + b_{k} \left[ \frac{||e_{i}||^{2}+||e_{j}||^{2}-||e_{k}||^{2}}{||e_{i}||^{2}} \right] \right\} \dots (5.15) \end{split}$$

where  $\|e_i\|$  is the length of the edge opposite vertex i, and  $b_i$ ,  $b_j$ ,  $b_k$  are the area coordinates of the point (x,y) with respect to triangle T. Area coordinates are coordinates that describe the position of a point within the interior of a triangle relative to the vertices of the triangle. The coordinates are based solely on the geometry of the triangle. Area coordinates are sometimes called "barycentric coordinates." The relative magnitude of the coordinates corresponds to area ratios, as shown in Figure 5.6

The xy coordinates of the interior point can be written in terms of the xy coordinates of the vertices using the area coordinates as follows:

$$x = b_i x_i + b_j x_j + b_k x_k...$$
 (5.16)

$$y = b_i y_i + b_j y_j + b_k y_k...$$
 (5.17)

$$1.0 = b_i + b_j + b_k....(5.18)$$

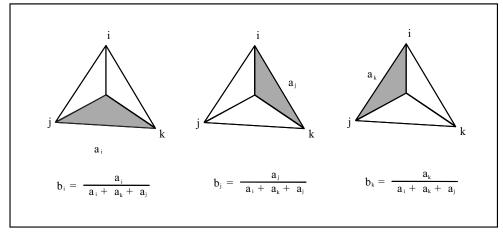


Figure 5.6 Barycentric Coordinate for a Point in a Triangle.

Solving the above equations for  $b_i$ ,  $b_i$ , and  $b_k$  yields:

$$b_{i} = \frac{1}{2A} \left[ (x_{j}y_{k} - x_{k}y_{j}) + (y_{j} - y_{k})x + (x_{k} - x_{j})y \right] ... (5.19)$$

$$b_{j} = \frac{1}{2A} \left[ (x_{k}y_{i} - x_{i}y_{k}) + (y_{k} - y_{i})x + (x_{i} - x_{k})y \right] ... (5.20)$$

$$b_k = \frac{1}{2A} \left[ (x_i y_j - x_j y_i) + (y_i - y_j) x + (x_j - x_i) y \right] \dots (5.21)$$

$$A = \frac{1}{2}(x_iy_j + x_jy_k + x_ky_i - y_ix_j - y_jx_k - y_kx_i) .....(5.22)$$

Using the weight functions defined above, the interpolating surface at points inside a triangle is computed as:

$$F(x,y) = W_i(x,y)Q_i(x,y) + W_i(x,y)Q_i(x,y) + W_k(x,y)Q_k(x,y) \dots (5.23)$$

where  $W_i$ ,  $W_j$ , and  $W_k$  are the weight functions and  $Q_i$ ,  $Q_j$ , and  $Q_k$  are the nodal functions for the three vertices of the triangle.

IDW interpolation using local weights as defined in equation 5.15 can be selected in the *Weighting Method* section of the *IDW Interpolation Options* dialog. This type of weighting is significantly faster than the standard weighting method, particularly if the number of scatter points is large.

## 5.7.5 Clough-Tocher Interpolation

The *Clough-Tocher* interpolation technique is often referred to in the literature as a finite element method because it has origins in the finite element method of numerical analysis. Before any points are interpolated, the scatter points are first triangulated to form a temporary TIN. A bivariate polynomial is defined over each triangle, creating a surface made up of a series of triangular Clough-Tocher surface patches.

The Clough-Tocher patch is a cubic polynomial defined by twelve parameters shown in Figure 5.7: the function values, f, and the first derivatives,  $f_x$  &  $f_y$ , at each vertex, and the normal derivatives,  $\partial f/\partial n$ , at the midpoint of the three edges in the triangle (Clough & Tocher, 1965; Lancaster & Salkauskas, 1986). The first derivatives at the vertices are estimated using the average slopes of the surrounding triangles. The element is partitioned into three subelements along seams defined by the centroid and the vertices of the triangle.

A complete cubic polynomial of the form:

$$F(x,y) = \sum_{j=0}^{3-i} c_{ij} x^i y^j \sum_{j=0}^{3-i} c_{ij} x^i y^j \qquad (5.24)$$

is created over each subtriangle with slope continuity across the seams and across the boundaries of the triangle. Second derivative continuity is not maintained across the seams of the triangle.

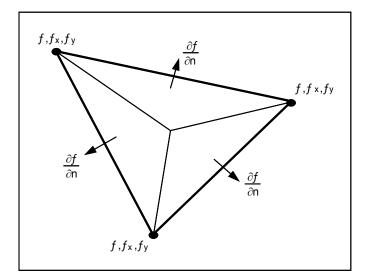


Figure 5.7 The Twelve Parameters Used to Define the Clough Tocher Triangle.

The form of equation 5.24 implemented in *SMS* is highly complex and is not included in this reference manual. The complete set of equations can be found in Jones (1990).

Since the Clough-Tocher scheme is a local scheme, it has the advantage of speed. Even very large scatter point sets can be interpolated very quickly. It also tends to give a very smooth interpolating surface which brings out local trends in the data set quite accurately.

Since a triangular network only covers the convex hull of a scatter point set, extrapolation beyond the convex hull is not possible with the Clough-Tocher interpolation scheme. Any points outside the convex hull of the scatter point set are assigned the *default extrapolation value* entered at the bottom of the *Interpolation Options* dialog.

# 5.7.6 Natural Neighbor Interpolation

*Natural neighbor* interpolation is also supported in *SMS*. Natural neighbor interpolation has many positive features. It can be used for both interpolation and extrapolation, and it behaves very well with clustered scatter points. Natural neighbor interpolation was first introduced by Sibson (1981). A more detailed description of natural neighbor interpolation in multiple dimensions can be found in Owen (1992).

The basic equation used in natural neighbor interpolation is identical to the one used in IDW interpolation (equation 5.11). As with IDW interpolation, the nodal functions can be either constants, gradient planes, or quadratics. The nodal function can be selected using the *Natural Neighbor Options* dialog (see Figure 5.8). The difference between the IDW interpolation and natural neighbor interpolation is the method used to compute the weights and the method used to select the subset of scatter points used for interpolation.

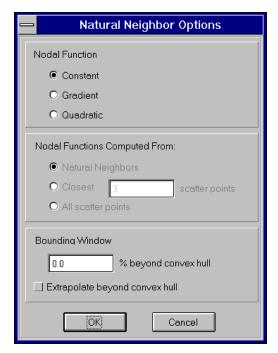


Figure 5.8 The Natural Neighbor Options Dialog.

Natural neighbor interpolation is based on the Thiessen polygon network of the scatter point set. The Thiessen polygon network can be constructed from the Delauney triangulation of a scatter point set (see Figure 5.9). A Delauney triangulation is a triangulated irregular network that has been constructed so that the Delauney criterion has been satisfied.

There is one Thiessen polygon in the network for each scatter point. The polygon for a scatter point encloses the area that is closer to the scatter point than any other scatter point. The polygons in the interior of the scatter point set are closed polygons and the polygons on the convex hull of the set are open polygons.

Each Thiessen polygon is constructed using the circumcircles of the triangles resulting from a Delauney triangulation of the scatter points. The vertices of the Thiessen polygons correspond to the centroids of the circumcircles of the triangles.

#### **Local Coordinates**

The weights used in natural neighbor interpolation are based on the concept of local coordinates. Local coordinates define the "neighborliness" or amount of influence any scatter point will have on the computed value at the interpolation point. This neighborliness is entirely dependent on the area of influence or Thiessen polygons of the surrounding scatter points.

To define the local coordinates for the interpolation point,  $P_n$ , the area of all Thiessen polygons in the network must be known. Temporarily inserting  $P_n$  into the TIN will cause the TIN and the corresponding Thiessen network to change, resulting in new Thiessen areas for the polygons in the neighborhood of  $P_n$ .

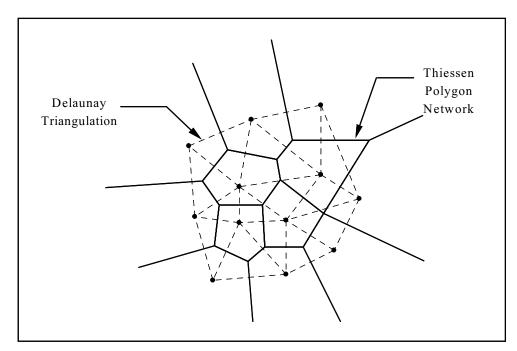


Figure 5.9 Delauney Triangulation and Corresponding Thiessen Polygon Network.

The concept of local coordinates is shown graphically in Figure 5.10. Points 1-10 are scatter points and  $P_n$  is a point where some value associated with points 1-10 is to be interpolated. The dashed lines show the edges of the Thiessen network before  $P_n$  is temporarily inserted into the TIN and the solid lines show the edges of the Thiessen network after  $P_n$  is inserted.

Only those scatter points whose Thiessen polygons have been altered by the temporary insertion of  $P_n$  are included in the subset of scatter points used to interpolate a value at  $P_n$ . In this case, only points 1, 4, 5, 6, & 9 are used. The local coordinate for each of these points with respect to  $P_n$  is defined as the area shared by the Thiessen polygon defined by point  $P_n$  and the Thiessen polygon defined by each point before point  $P_n$  was added. The greater the common area, the larger the resulting local coordinate, and the larger the influence or weight the scatter point has on the interpolated value at  $P_n$ .

If we define  $\kappa(n)$  as the Thiessen polygon area of  $P_n$  and  $\kappa_m(n)$  as the difference in the Thiessen polygon area of a neighboring scatter point,  $P_m$ , before and after  $P_n$  is inserted, then the local coordinate  $\lambda_m(n)$  is defined as:

$$\lambda_{m}(n) = \frac{\kappa_{m}(n)}{\kappa(n)} \qquad (5.25)$$

The local coordinate  $\lambda_m(n)$  varies between zero and unity. If  $P_n$  is at precisely the same location as  $P_m$ , then the Thiessen polygon areas for  $P_n$  and  $P_m$  are identical

and  $\lambda_m(n)$  has a value of unity. In general, the greater the relative distance  $P_m$  is from  $P_n$ , the smaller its influence on the final interpolated value.

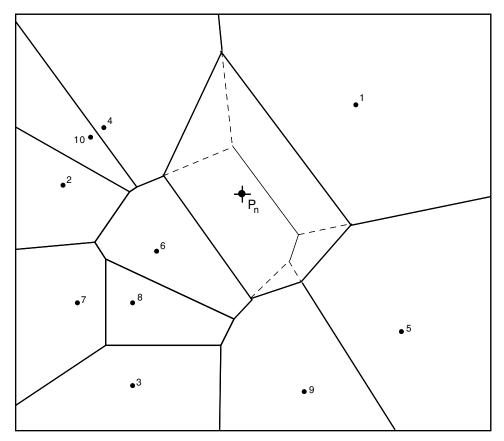


Figure 5.10 Overlapping Thiessen Polygon Areas Used to Compute Local Coordinates.

The weights used in natural neighbor interpolation are computed by normalizing the local coordinates so that they sum to one:

$$w_{m}(n) = \frac{\lambda_{m}(n)}{\sum_{i=1}^{p} \lambda_{i}(n)}$$
 (5.26)

where  $w_m(n)$  is the weight of scatter point  $P_m$  with respect to the interpolation point  $P_n$ , and p is the number of points in the neighborhood of  $P_n$  with non-zero local coordinates.

### **Bounding Window**

As shown in Figure 5.9, the Thiessen polygons for scatter points on the perimeter of the TIN are open-ended polygons. Since such polygons have an infinite area, they cannot be used directly for natural neighbor interpolation. In order to make the area

of these polygons finite, a bounding box or window is superimposed on the scatter point set (see Figure 5.11). Polygons for points on the exterior of the scatter point set are clipped or truncated to the bounding window.

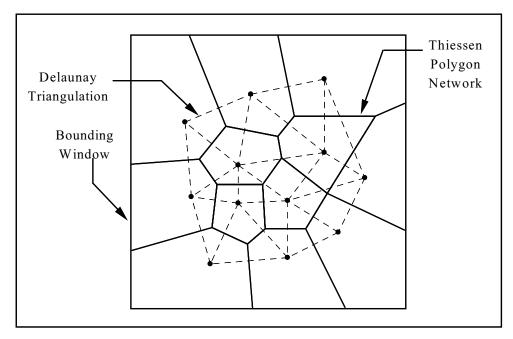


Figure 5.11 Bounding Window Used in Natural Neighbor Interpolation.

With finite Thiessen polygon areas on the perimeter of the scatter point set, it is possible to perform extrapolation (estimate values for interpolation points outside the convex hull of the scatter point set) as well as interpolation. However, the values computed by extrapolation are somewhat influenced by the relative size of the bounding window with respect to the size of the scatter point set. The larger the bounding window, the greater the influence of the perimeter scatter points on the extrapolated values. If the bounding window is extremely large, the extrapolated values will be influenced only by the points on the convex hull of the scatter point set. If the bounding window is not significantly larger than the scatter point set, the extrapolated values will be influenced by interior scatter points in the neighborhood of the interpolation point in addition to perimeter scatter points near the interpolation point. The relative size of the bounding window can be altered using the *Natural Neighbor Interpolation Options* dialog. In addition, the dialog can be used to turn off the extrapolation option entirely.

CHAPTER 6

# Map Module 🕾

The *Map* module provides a suite of tools for defining conceptual models in a GIS format, adding annotation to a plot, displaying digital background maps, and displaying CAD drawings.

Four types of objects are supported in the *Map* module: feature objects, drawing objects, digital images, and DXF files.

Feature objects are used to provide GIS capabilities within *SMS*. Feature objects include points, arcs, and polygons. Feature objects can be grouped into layers or coverages. A set of coverages can be constructed representing a conceptual model of a *surface water* modeling problem. Feature objects can also be used for automated mesh generation.

Drawing objects provide a simple method for adding annotation to a plot. Drawing objects include text, arrows, lines, rectangles, and ovals.

Images are scanned maps or aerial photos imported from TIFF files. Images are displayed in the background for on-screen digitizing or model placement or simply to enhance the display of a model.

DXF files are CAD drawings which can be imported into *SMS* and displayed in the *Graphics Window* to assist in model placement or simply to enhance the display of a model. Polylines from the imported DXF files can also be converted into feature objects for use in mesh generation.

# 6.1 Feature Objects

Feature objects in *SMS* have been patterned after Geographic Information Systems (GIS) objects and include points, nodes, arcs, and polygons. Feature objects can be grouped together into coverages, each coverage defining a particular set of information. The primary use of feature objects is to construct 2D finite element meshes. The mesh domain can be described with polygons, each polygon representing a material zone. Special refine points can also be identified in the interior of the domain. Boundary parameters such as flow and head values can also be identified. A mesh can then be automatically generated from the polygons and the points by filling in the interior of the polygonal zones with nodes and elements.

# 6.1.1 Feature Object Types

The definition of feature objects in *SMS* follows that used by typical GIS software that supports vector data. The basic object types are points, nodes, vertices, arcs, and polygons. The relationship between these objects is illustrated in Figure 6.6.1.

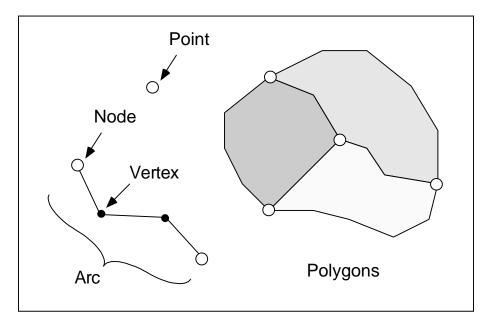


Figure 6.6.1 Feature Object Types.

#### **Points**

Points are XY locations that are not attached to an arc. Points have unique ids and can be assigned attributes. Points are often used to refine a mesh in an area of

interest. Points are also used when importing a set of XY locations for the purpose of creating arcs or polygons.

#### Arcs

Arcs are sequences of line segments or edges which are grouped together as a single "polyline" entity. Arcs have unique ids and can be assigned attributes. Arcs are grouped together to form polygons or are used independently to represent linear features such as specified head. The two end points of an arc are called "nodes" and the intermediate points are called "vertices".

#### **Nodes**

Nodes define the beginning and ending XY locations of an arc. Nodes have unique ids and can be assigned attributes.

#### **Vertices**

Vertices are XY locations along arcs in between the beginning and ending nodes. They are used solely to define the geometry of the arcs. Vertices do not have ids or attributes.

## **Polygons**

Polygons are a group of connected arcs that form a closed loop. A polygon can consist of a single arc or multiple arcs. If two polygons are adjacent, the arc(s) forming the boundary between the polygons is shared (not duplicated).

Polygons may not overlap. However, a polygon can have a hole(s) defined by having a set of closed arcs defining interior polygon(s). An example of such a case is shown in Figure 6.2 where three arcs are used to define two polygons. Polygon A is made up of arcs 1, 2, 3 and 4, whereas polygon B is defined by a single arc (arc 2). For polygon A arcs 1, 3, and 4 define the exterior boundary whereas arc 2 defines a hole.

Polygons have unique ids and can be assigned attributes. Polygons are used to represent material zones such as main channel, overbank, flood plain, lakes, etc.

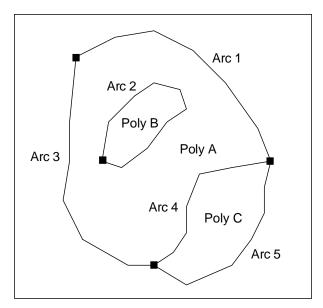


Figure 6.2 Polygon With Holes.

## 6.1.2 Feature Object Tools

Several tools are provided in the *Tool Palette* for creating and editing feature objects. These tools are located in the dynamic portion of the *Tool Palette* and are only available when the *Map* module is active. The tools are as follows:

## Select Point/Node

The *Select Point/Node* tool is used to select existing points or nodes. A selected point/node can be deleted, moved to a new location, or operated on by one of the commands in the *Feature Objects* menu. The coordinates of selected points/nodes can be edited using the *Edit Window*. Double clicking on a point or node with this tool brings up the *Point* or *Node Attribute* dialog.

# Select Vertex

The *Select Vertex* tool is used to select vertices on an arc. Once selected, a vertex can be deleted, moved to a new location, or operated on by one of the commands in the *Feature Objects* menu. The coordinates of a selected vertex can be edited using the *Edit Window*.

## Select Arc

The *Select Arc* tool is used to select arcs for operations such as deletion, redistribution of vertices, or building polygons. Double clicking on an arc with this tool brings up the *Arc Attributes* dialog.

## Create Point

The *Create Point* tool is used to interactively create new points. A new point is created for each location the cursor is clicked on in the *Graphics Window*. Once the point is created, it can be repositioned or otherwise edited with the *Select Point/Node* tool.

# Create Vertex

The *Create Vertex* tool is used to interactively create new vertices along an existing arc. This is typically done to add more detail to the arc. A new vertex is created for each location the cursor is clicked on in the *Graphics Window*, that it is within a given pixel tolerance of an existing arc. Once the vertex is created, it can be repositioned with the *Select Vertex* tool.

## Create Arc

The *Create Arc* tool is used to interactively create new arcs. An arc is created by clicking once on the location where the arc is to begin, clicking once to define the location of each of the vertices in the interior of the arc, and double-clicking at the location of the end node of the arc.

As arcs are created, it is often necessary for the beginning or ending node of the arc to coincide with an existing node. If you click on an existing node or point (within a given pixel tolerance) when beginning or ending an arc, that node is used to define the arc node as opposed to creating a new node. Also, if you click on a vertex or segment of another arc while creating an arc, that vertex is converted to a node or a new node is created and the node is used in the new arc.

While creating an arc, it is not uncommon to make a mistake by clicking on the wrong location. In such cases, hitting the *BACKSPACE* key backs up the arc by one vertex. The *ESCAPE* key can also be used to abort the entire arc creation process at any time.

# Select Polygon

The *Select Polygon* tool is used to select previously created polygons for operations such as deletion, assigning attributes, etc. A polygon is selected by clicking anywhere in the interior of the polygon. Double-clicking on a polygon with this tool brings up the *Polygon Attributes* dialog.

## 6.1.3 Build Polygon

While the other feature objects can be constructed with tools in the *Tool Palette*, polygons are constructed with the *Build Polygon* command. Since polygons are

defined by arcs, the first step in constructing a polygon is to create the arcs forming the boundary of the polygon. Once the arcs are created, they should be selected with the *Select Arc* tool, and the *Build Polygon* command should be selected from the *Feature Objects* menu.

The *Build Polygon* command can be used to construct one polygon at a time or to construct several polygons at once. If the selected arcs form a single loop, only one polygon is created. If the arcs form multiple loops, a polygon is created for each unique (non-overlapping) loop. If no arcs are selected, all of the currently defined polygons in the active coverage are used to create polygons.

#### 6.1.4 Clean

The *Clean* command is used to fix errors in feature object data. Specifically, it prompts for a snapping tolerance and minimum dangling arc length, and then uses these parameters to do the following:

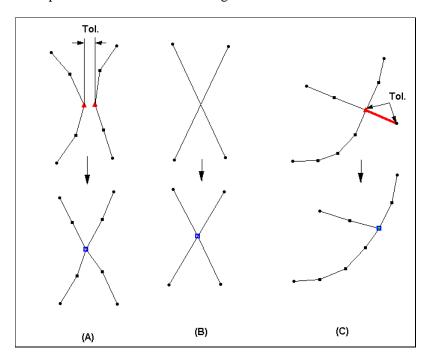


Figure 6.3 Cleaning Arcs. (A) snapping (B) intersecting (C) dangling arcs

- 1. A check is made to see if any nodes are within the specified tolerance of other nodes. If so, the nodes are snapped together. (see Figure 6.3 A)
- 2. A check is made to see if any arcs intersect. If so, a node is created at the intersection and the arcs are split. (see Figure B)
- 3. A check is made for dangling arcs (arcs with one end not connected to another arc) with a minimum length. If any are found they are deleted. (see Figure 6.3 C)

All objects in the active coverage are "cleaned".

#### 6.1.5 Vertex <-> Node

In some cases, it is necessary to split an arc into two arcs. This can be accomplished using the *Vertex* <-> *Node* command. Before selecting this command, a vertex on the arc at the location where the arc is to be split should be selected. The selected vertex is converted to a node and the arc is split in two.

The *Vertex* <-> *Node* command can also be used to combine two adjacent arcs into a single arc. This is accomplished by converting the node joining the two arcs into a vertex. Two arcs can only be merged if no other arcs are connected to the node separating the arcs. Otherwise, the node must be preserved to define the junction between the branching arcs.

#### 6.1.6 Redistribute Vertices

The primary function of the vertices of an arc is to define the geometry of the arc. If the arcs are to be used for automatic mesh generation, the spacing of the vertices is important. The spacing of the vertices defines the density of the elements in the resulting mesh. Each edge defined by a pair of vertices becomes the edge of an element. The mesh gradation is controlled by defining closely spaced vertices in regions where the mesh is to be dense and widely spaced vertices in regions where the mesh is to be coarse.

When spacing vertices along arcs, the *Redistribute* command in the *Feature Objects* menu can be used to automatically create a new set of vertices along a selected set of arcs at either a higher or lower density. The desired arc should be selected prior to selecting the *Redistribute* command. The *Redistribute* command brings up the dialog shown in Figure 6.4.

The current status of the selected arc is given at the top of the dialog. This includes the number of segments and spacing of those segments. When multiple arcs are selected, the current status is a combination of all selected arcs. However, the parameters set in this dialog apply to each arc individually. Therefore if multiple arcs were selected, each arc would reflect the options selected in this dialog.

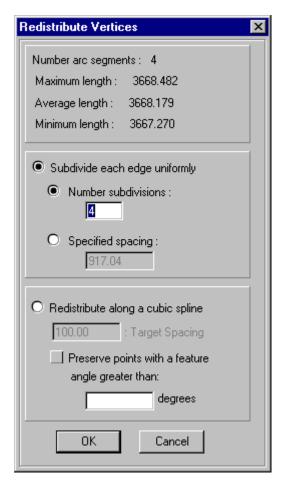


Figure 6.4 Redistribute Vertices Dialog.

The following options are available for redistributing vertices:

### **Linear Interpolation**

If the *Linear interpolation* options is specified, then either a number of intervals or a target spacing can be given to determine how points are redistributed along the selected arcs. In either case, the new vertices are positioned along a linear interpolation of the original arc. The arc may change shape due to the fact that original vertices are removed as the new vertices are created. This may round corners from the arc.

#### Spline Interpolation

If the *Spline interpolation* option is specified, vertices are redistributed along a series of cubic splines defined by the original vertices of the selected arcs.

The difference between the linear and spline interpolation methods is illustrated in Figure 6.5.

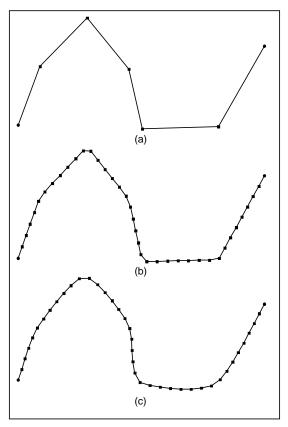


Figure 6.5 Redistributing Vertices. (a) Original Arc (b) Linear Interpolation (c) Spline Interpolation.

## 6.1.7 Coverages

As mentioned before, feature objects are grouped into coverages. A coverage is similar to a layer in a CAD drawing. Each coverage represents a particular set of information. For example, one coverage could be used to define one dimensional model features such as river centerlines and cross-section locations. Another coverage could be used to define two dimensional material zones, or to do observations (next release). These objects could not be included in a single coverage since polygons within a coverage are not allowed to overlap and material regions will cover one dimensional geometry features.

When *SMS* is first launched, a default coverage is created. Any feature objects created are added to this coverage. When multiple coverages are created, one coverage is designated the "active" coverage. New feature objects are always added to the active coverage and only objects in the active coverage can be edited.

Each coverage has a coverage type associated with it. The coverage type controls what attributes are assigned to the objects within the coverage. For example, with a 2D Mesh type of coverage, materials can be assigned to polygons and element sizes can be assigned to points.

Options related to coverages are controlled with the *Coverages* dialog (Figure 6.6). This dialog is accessed through the *Coverages* command in the *Feature Objects* menu.

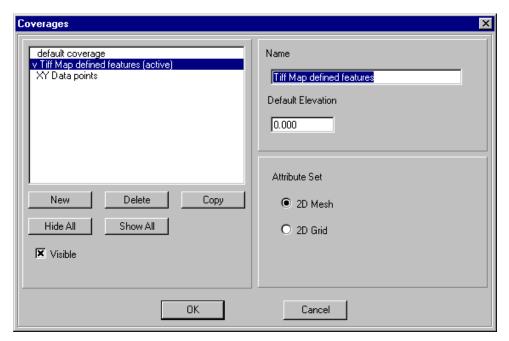


Figure 6.6 The Coverages Dialog.

The currently defined coverages are listed in the text box in the upper left corner of the dialog. One of the coverages in the list is always highlighted. The name and other attributes associated with the highlighted coverage are edited with the other fields in the dialog. The coverage that is highlighted when the *OK* button is selected is the active coverage.

#### Name

Each coverage has a name that is used in the list to identify the coverage. The name of a coverage is edited by selecting the coverage and editing the name in the *Name* field.

#### **Default Elevation**

Coverages are essentially two-dimensional entities. All objects within a coverage are displayed in the same XY plane. The default elevation field can be used to define the Z elevation of the XY plane containing the coverage.

#### **Creating/Deleting Coverages**

A new coverage is created by selecting the *New* button. This adds a new empty coverage to the list. Another way to create a new coverage is with the *Copy* button. This creates a new coverage with the same set of feature objects as the selected coverage. This is useful when two coverages share the same boundary or zones as

previously discussed. An existing coverage can be deleted by highlighting the coverage and selecting the *Delete* button.

## **Display Color**

When several coverages are present, the display of coverages can become confusing. Each of the feature objects in a coverage has a set of display options (color, line style, etc.) that can be edited in the *Display Options* dialog. However, these colors are only used to display the objects in the active coverage. All of the objects in the inactive coverages are displayed using the same color. By default, this color is a light gray color. The inactive coverage color can be edited in the *Display Options* dialog.

#### Visibility

In some cases it is useful to hide some or all of the coverages. Each coverage has a visible flag that can be edited. Only visible coverages are displayed. The flag for a coverage is edited by highlighting the coverage and selecting the visible toggle in the lower left corner of the *Coverages* dialog. Coverages which are visible have a small "v" displayed before the coverage name. The visible flag for all coverages can be edited at once using the *Hide All* and *Show All* buttons.

#### **Attributes Sets**

Each coverage is assigned a coverage type which controls which set of attributes are associated with the coverage. The appropriate attribute set for a coverage depends on the intended used of the coverage.

Currently *SMS* only supports 2D Mesh attributes. This coverage type is used when constructing a 2D mesh from feature objects. The attributes and commands associated with this process are described in detail beginning on page 6-21.

Work is underway to support attribute sets for observation and model verification, 1d cross section and profile tools for backwater computation and 2d boundary fitted finite difference grids.

## 6.1.8 Display Options

The *Display Options* command in the *Feature Objects* menu can be used to control the display of coverages and feature objects. The options that appear in the *Display Options* dialog depend on the type of the active coverage. A sample dialog is shown in Figure 6.7.

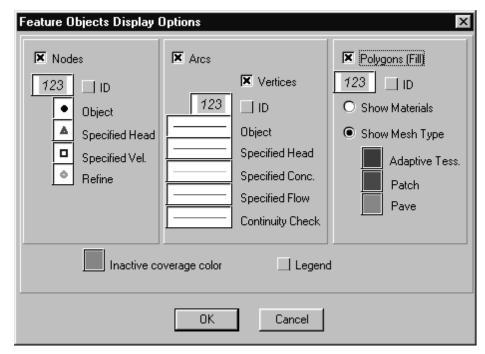


Figure 6.7 The Feature Object Display Options Dialog.

#### ID

If this option is selected, the id of each of the feature objects is displayed next to the object. The graphical attributes of the text used to display the ids are edited by clicking on the fields on the left side of each id toggle box. ID's are only displayed if the corresponding feature object is displayed.

#### **Nodes**

This option is used to display nodes. The graphical attributes of the nodes (symbol, color, size, etc.) are edited by clicking on the fields in the node section of the dialog.

#### **Arcs**

This option is used to display arcs. The graphical attributes of the arcs (color, line style, thickness, etc.) are edited by clicking on the fields in the arc section of the dialog.

#### **Vertices**

This option is used to display the vertices of arcs. A small dot is placed on the arcs at the location of each of the vertices. The color of the vertices is the same as the color of arcs. Vertices are only displayed when arcs are displayed.

## **Polygons**

If this option is selected, polygons are displayed filled. The graphical attributes of the polygons (fill color) are edited using the fields in the polygon section of the dialog. The polygon fill color may represent either the mesh type or material property.

#### **Inactive Color**

The inactive color is used to display all of the objects in inactive coverages.

### Legend

The legend item can be used to display a legend listing each of the feature object types being displayed. The legend shows what graphical attributes (symbol, line style, fill color and pattern) are being used to display each feature object type.

#### Summary:

Polygons, arcs and isolated points may all have only one type of attribute. Nodes, however, may be connected to multiple arcs of different types. Because of this, nodes may have more than one type of attribute. Nodes are limited though, to having only the same types of attributes that the attached arcs have.

## 6.1.9 Assigning and Editing Attributes

Attributes of feature objects are assigned and edited by selecting the objects and selecting the *Attributes* command from the *Feature Objects* menu. Depending on the type of the object selected, this brings up the *Point/Node Attributes* dialog, the *Arc Attributes* dialog, or the *Polygon Attributes* dialog. These dialogs can also be accessed by double clicking on a feature object. The contents of the dialogs (the attribute options) depend on the attribute set assigned to the active coverage. Creating a new coverage or assigning an active coverage is described in Section 6.1.7

## 6.2 2D Mesh Coverage

#### 6.2.1 Point/Node Attributes

The Point/Node Attributes dialog for 2D Mesh coverage is shown in Figure 6.8.

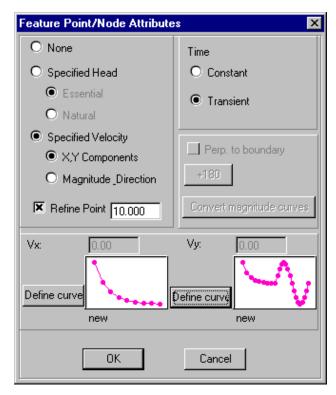


Figure 6.8. The Point/Node Attributes Dialog.

This dialog is used to input and edit the attributes assigned to both nodes and points. The main portion of this dialog is the upper left area. The parameters for the attribute types are viewed and edited in this portion of the dialog.

Nodes or points can be assigned either constant (steady state) or transient (time-dependent) conditions. To assign constant conditions to the selected node or point, select the *Constant* radio button in the *Time* section of the dialog and enter the value into the available edit field(s) at the bottom of the dialog. To assign transient boundary conditions to the node, select the *Transient* radio button and click the *Define Curve* button. For transient head boundary conditions, only the left curve is active, representing the changing head. For transient velocity boundary conditions, both curves are active. The velocity can be entered as a combination of magnitude and direction, or in terms of its *x* and *y* components. Clicking on a *Define Curve* button will bring up the *XY Series Editor* dialog described in Chapter 9. This editor allows the user to define a function of values -vs- time. Values must be defined for transient concentrations before selecting *OK* in this dialog.

The attributes assigned to feature points and nodes will be saved as nodal boundary conditions when the feature objects are converted to a two dimensional mesh (Section 6.2.3). The attributes that can be assigned to feature nodes and points are as follows:

#### **Specified Head**

To assign head boundary conditions to the selected node or point, select the *Specified Head* radio button and enter the appropriate value or define the curve. A node/point

can also be assigned to be essential or natural by clicking on the appropriate radio button. These boundary conditions are used in the FESWMS (chapter 11) and RMA2 (chapter 8) analysis packages.

### **Specified Velocity**

To assign velocity boundary conditions to the selected node or point, select the *Specified Velocity* radio button. Velocity boundary conditions may be defined as either vector components in the x and y directions, or as a magnitude and direction. If using the magnitude and direction method, selecting the *Perp. to boundary* button will calculate and angle which is perpendicular to the boundary at the selected node. Clicking the +180 button will add 180 degrees to the transient angle(s). The angle is measured in degrees from the positive x-axis to the direction in which the vector points. Since velocity boundary conditions are saved as vector components in the x and y directions, any constant velocity boundary conditions defined as magnitude and direction are automatically converted into vector components. To convert dynamic magnitude and direction curves into vector component curves, click the *Convert magnitude curves* button.

#### **Refine Point**

In addition to the attributes listed in the radio group, a point can be made a refine point by clicking on the Refine Point button. A refine attribute is assigned to points or nodes to automatically increase the grid density around a point when a mesh is constructed. The element size can be entered into the edit field to the right of the button.

## 6.2.2 Arc Attributes Dialog

The *Arc Attributes* dialog is shown in Figure 6.9. This dialog is used to input and edit attributes assigned to arcs. The most significant function of this dialog is to specify which type of attribute is assigned to the arc.

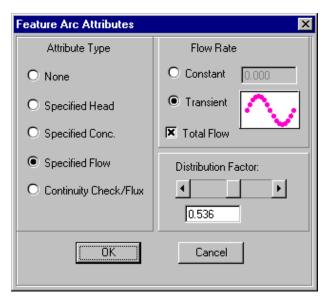


Figure 6.9. The Arc Attributes Dialog

The different types of attributes which can be assigned to an arc are on the left side of the dialog. The parameters for each attribute type are edited on the right side of the dialog.

Specified head, flow, or concentration may be either constant or transient. For a constant value, a single value is entered in the edit field. Transient values are displayed as a curve in a small window. Clicking in this window brings up the *XY Series Editor* described in Chapter 13. Before any XY series has been entered for a particular parameter, this window will show "Define Series". Values must be defined for transient concentrations before selecting *OK* in this dialog.

The feature arcs will be converted to nodestrings when the feature objects are converted to a two dimensional mesh (Section 6.2.3). The attributes assigned to feature arcs will be saved as either *RMA2* or *FESWMS* nodestring boundary conditions after the meshing takes place.

The attributes that can be assigned to feature arcs are as follows:

#### Specified Head

To assign head boundary conditions to the selected arc, select the *Specified Head* radio button and enter the appropriate value or define the curve. An arc can also be assigned to be essential or natural by clicking on the appropriate radio button in the lower right section of the dialog. These boundary conditions are used in *FESWMS* 

and will be assigned to a mesh nodestring after meshing (Section 6.2.3). If an arc is assigned an essential boundary condition, *FESWMS* will not allow the water surface elevation to fluctuate at all. *FESWMS* allows small water surface elevation fluctuations if a natural boundary condition is assigned. All specified heads in RMA2 and HIVEL are assumed to be natural.

#### **Specified Concentration**

To assign specified concentration boundary conditions to the selected arc, select the *Specified Conc*. radio button. This concentration corresponds to the sediment concentration for nodestrings in *SED2D-WES*.

#### **Specified Flow**

To assign flow boundary conditions to the selected arc, select the *Specified Flow* radio button and enter the appropriate values or define the curve. In addition to the flow rate, the user must specify a distribution factor. The distribution factor is used in *RMA2* and allows the flow to be distributed by depth rather than equally along the length of the nodestring after meshing. The user specifies a value using a scroll bar. The ends of the scroll bar correspond to the two extremes, length distribution and depth distribution. Values in the middle blend the two methods.

## **Continuity Check/Flux**

To assign an arc to be a flux string, select the *Continuity Check/Flux* radio button. A flux string is a nodestring at which continuity checks will be made during a *FESWMS* or RMA2 solution run. Flux strings are generally placed at cross sections of the model.

### **Total Flow**

In addition to the attributes listed in the radio group, an arc can be made a total flow nodestring if the arc is assigned to be a flux string. Note: SMS implicitly assigns a nodestring to be a flux string if it is a specified head, concentration, or flow string.

## 6.2.3 Polygon Attributes Dialog

The *Polygon Attributes* dialog is shown in Figure 6.10.

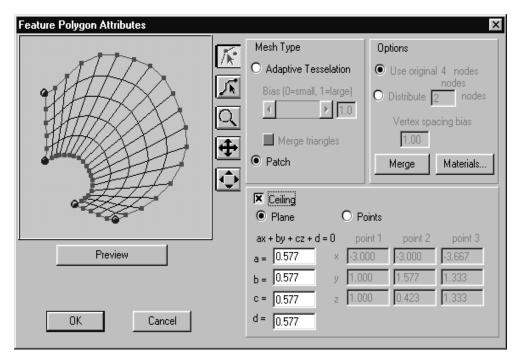


Figure 6.10. The Polygon Attributes Dialog

This dialog is used to set the attributes for feature polygons. The main portion of this dialog is the upper middle area. A polygon can be assigned a mesh type in this area.

### Mesh Type

There are two mesh types that can be assigned to each polygon. They are as follows:

#### Adaptive Tessellation

A polygon can be assigned a mesh type of Adaptive Tessellation by clicking on the *Adaptive Tessellation* button. Adaptive Tessellation is a meshing technique described in Section 6.2.6.

The *Bias* can be used to indicate whether the meshing algorithm should favor the creation of large or small elements. In either case, the elements in the interior of the mesh will honor the arc edges and the element sizes specified at nodes. The bias simply controls the element sizes in the transition region. If the large bias is chosen, the elements transition more quickly to the larger sizes when moving away from an arc with short edge lengths. If a small bias is chosen, the elements transition more slowly from small to large.

If the *Merge After Meshing* option is used, *SMS* merges a portion of the triangular elements into quadrilateral elements following the meshing process. This tends to reduce the total number of elements in the mesh. This option is only used for previewing a meshed polygon. The elements can be merged after the meshing process is completed using the *Merge Elements* command in the *2D Mesh* module.

This command and the element merging process are described in more detail in Section 6.2.6.

#### Patch

Patching is a meshing process described in Section 6.2.6. Patches must have three or four sides, with an equal number of segments on opposing sides. Both triangular and quadrilateral polygons can be meshed from this dialog. If a polygon cannot be patched, a help string under the *preview window* explains what needs to be changed. Modifying an existing polygon to have three or four sides is described in the following section.

#### **Options**

In the *Options* section of the dialog, there are several tools for modifying the existing polygon. They are as follows:

### Merge/Split

A patch must have three or four sides to be valid. The *Merge/Split* option allows the user to merge two or more arcs to increase/reduce the number of sides. Arcs can be merged by selecting the node connecting the two arcs with the *Select Node* tool and choosing the *Merge* button. Nodes can be distinguished from vertices by a small circle around them. In order to split two arcs, select the node connecting the two arcs and choose the *Merge/Split* button. The button will display *Split* and the node between the two arcs will be displayed red when the arcs are merged. The button will display *Merge* and the node will be displayed blue when the arcs are unmerged.

#### Distribute Nodes

An arc can be selected using the *Select Arc* tool under the *preview window* and the number of segments can be changed. This is useful in creating a patch since opposing patch segments must have equal sides. The original number of segments can be used by clicking on the *Use Original Nodes* button. By clicking on the *Distribute Nodes* button, the user can specify the number of segments. The spacing between the segments can be set in the *Vertex spacing bias* edit field. The value of the bias specifies the proportional distance between the last two nodes compared to the distance between the first two nodes. For example, if the bias is 2.0, the distance between the last two nodes. The bias spacing can be reversed by entering the inverse of the existing bias.

#### Materials

The material assigned to each polygon can be set by choosing the *Materials*... button in the *Options* area of the dialog. This brings up the Materials Editor dialog

described in Section 2.8.4. The material assigned to the polygon will be assigned to the corresponding mesh element after meshing (Section 6.2.6).

### Ceiling

A polygon can be assigned a ceiling value, or a value above which water flow is prohibited. The ceiling will be assigned to each mesh node after meshing has taken place (Section 6.2.6). Node ceilings are used in *FESWMS* (Section 11.11).

To assign a ceiling value, the user can either enter four points or the a, b, c, and d values of a plane equation. The ceiling value for each node will be interpolated and assigned after meshing.

#### **Preview**

To preview what the polygon will look like after meshing, select the *preview* button.

## 6.2.4 Constructing 2D Meshes

Although several options are provided in the 2D Mesh module for automated mesh generation, the simplest and most powerful method for generating a mesh is to use feature objects and the Map -> 2D Mesh command. The process of constructing a mesh from feature objects is illustrated in Figure 6.11.

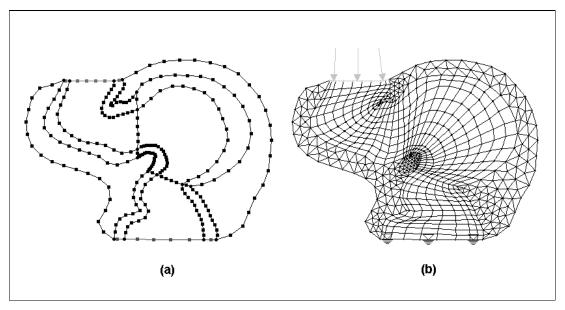


Figure 6.11 Mesh Generation with Feature Objects. (a) Feature Objects (b) Resulting Mesh.

The types of feature objects that are used to guide the construction of the mesh are shown in Figure 6.11a and the resulting mesh is shown in Figure 6.11b. The polygons shown in Figure 6.11a represent material zones and also define the domain of the region to be meshed. The spacing of the edges (defined by the arc vertices) of

arcs is used to guide the number and size of elements that are generated. One element is generated adjacent to each arc edge. The elements in the interior of the mesh domain are generated in a manner that gradually transitions the element sizes between arcs with small edges and arcs with large edges. This allows the user to control the mesh density and ensure that higher mesh density is used where necessary. Both arcs on polygons and arcs representing interior boundaries such as rivers are honored in the meshing process.

An element size can be assigned to individual points in the interior of the mesh. As the mesh is constructed, the element sizes are gradually transitioned so that the elements adjacent to the point have the specified size. In most cases, this element size is relatively small and is intended to cause the mesh to be refined about the point.

#### 6.2.5 2D Mesh Attributes

Attributes are assigned to feature objects by selecting the object(s) and selecting the *Attributes* command in the *Feature Objects* menu. Attributes can also be assigned by double clicking on an object. When the active coverage is a *2D Mesh* type coverage, attributes can be assigned to polygons and points.

#### **Polygons**

Polygons in a 2D Mesh type coverage can be assigned a material type. When the Attributes command is selected, the standard Materials dialog appears. This dialog is described on page 2-15. The dialog can be used to select a previously defined material, or to create a new material. The material that is highlighted in the materials list in the upper left corner of the dialog when the OK button is selected is the material that is assigned to the polygon. When the mesh is generated, all of the elements created inside the polygon are assigned the polygon's material type.

#### **Points**

Points in a 2D Mesh type coverage can be assigned an element size. When the Attributes command is selected, a dialog with a single edit field appears prompting the user for the element size. The size represents an edge length. When the mesh is generated, the elements just adjacent to the point are equilateral triangles with the designated edge length.

If an element size is not assigned to a point in a 2D Mesh coverage, the point will still be honored in the mesh. A node will be created at the exact location of the node. However, the sizes of the elements in the vicinity will depend on the transitioning required by other objects in the coverage.

#### 6.2.6 Map -> 2D Mesh

Once a set of 2D Mesh type feature objects has been created, the Map -> 2D Mesh command can be used to generate a 2D finite element mesh from the objects. When the Map -> 2D Mesh command is selected, the dialog shown in Figure 6.12 appears.

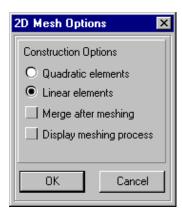


Figure 6.12 The 2D Mesh Options Dialog.

If at least one of the polygons being meshed is filled using the *Adaptive Tessellation* option in the *Feature Polygon Attributes* dialog, the user can specify if quadratic or linear elements will be created. Choosing the *Quadratic Elements* option will create either six node triangular or eight node quadrilateral elements. Choosing the *Linear Elements* option will create either three node triangular or four node quadrilateral elements.

If the *Merge After Meshing* option is used, *SMS* merges a portion of the triangular elements into quadrilateral elements following the meshing process. This tends to reduce the total number of elements in the mesh. The elements can also be merged after the meshing process is completed using the *Merge Elements* command in the *2D Mesh* module. This command and the element merging process are described in more detail on page 4-22.

If the *Display meshing progress* option is selected, each step of the meshing process is displayed. This tends to slow down the meshing significantly.

#### **Construction Method Options**

In the *Feature Polygon Attributes* dialog, there are two construction methods that specify the configuration of the elements that will fill the polygons. These options are:

#### Adaptive tessellation

When a polygon is filled using the *Adaptive Tessellation* option, *SMS* uses the existing node spacing on the boundary of the input polygon to determine the element sizes on the interior. If the input polygon has varying node densities along its perimeter, *SMS* attempts to create a smooth element size transition between these

areas of differing densities. By altering the size bias in the *Feature Polygon Attributes* dialog, the user can indicate whether *SMS* should favor the creation of large or small elements. Decreasing the bias will result in smaller elements; increasing the bias will result in larger elements. A sample mesh created using the adaptive tessellation technique is shown in Figure 6.13.

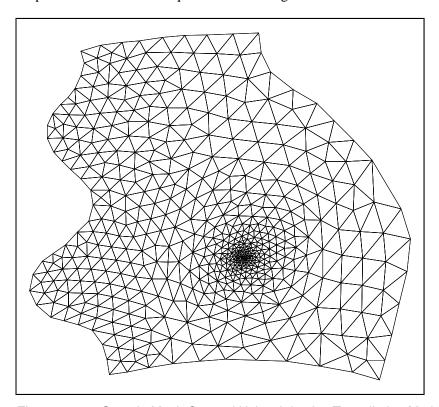


Figure 6.13 Sample Mesh Created Using Adaptive Tessellation Method.

#### **Patches**

Rectangular and triangular patches of elements can be created by creating the boundary of a region and filling the interior of the patch automatically. Patches can be created from existing mesh nodes as described in Sections 4.7.3 & 4.7.4 or from feature polygons. A feature polygon must consist of three or four sides. The modification of feature polygons in order to create a patch is described in Section 6.2.3.

## 6.3 Drawing Objects

The drawing objects in the *Map* module provide a set of tools for adding simple graphics and annotation to a plot. These tools are not intended to be a full-featured drawing package as would be found in products like *AutoCAD* or *Corel Draw*. However, they can be very useful for adding titles, arrows, and other annotation to a

plot so that the plot can be directly included in a project report without the need to import the plot into an external drawing package prior to report generation.

The types of drawing objects that can be created are text, lines (including arrows), rectangles, and ovals. Drawing objects are created and edited using tools in the *Tool Palette*. Drawing objects are saved in the Map file along with feature objects.

## 6.3.1 Drawing Object Tools

The following drawing object tools are in the dynamic portion of the *Tool Palette* when the *Map* module is activated. Only one tool is active at any given time.

## T Create Text Tool

The *Create Text* tool is used to create a single line text string. The location clicked on defines where the text string will be placed. After clicking on a location, the *Text Attributes* dialog appears allowing you to enter the text string and choose the font, color, etc.

# Create Rectangle

The *Create Rectangle* tool is used to create wire frame or filled rectangles. Rectangles can be used to represent buildings, frame a series of text strings, etc.. Rectangles are created with this tool by dragging a rectangle with the mouse at the location on the screen where you wish to place the rectangle.

## Create Oval

The *Create Oval* tool can be used to create wire frame or filled ovals. Ovals are created by dragging a rectangle with the mouse at the location on the screen where you wish to place the oval. The rectangle width and height determine the major and minor axes of the oval.

## Create Line

The *Create Line* tool can be used to create single line segments or polylines (a series of connected segments). An arrowhead can be placed on either end of the line. Lines are typically used in conjunction with text strings to highlight key features in a plot. A line is created by clicking on a series of points on the screen with the mouse and double-clicking to end to end the line. The color, line style, and arrowhead options of a line are edited with the *Attributes* dialog described below.

## B Select Drawing Objects

The *Select Drawing Objects* tool is used to select previously created text, rectangles, ovals, and lines. Once selected, a drawing object can be moved to another location by clicking on the object and dragging it to a new location. Lines, rectangles, and

ovals can be resized by dragging the handles that appear on the corners or ends of the object when the object is selected. The *Select Drawing Objects* tool is also used to edit the graphical attributes as described in the following section.

## 6.3.2 Display Attributes

Each type of drawing object has a set of graphical attributes that can be edited by selecting the object with the *Select Drawing Objects* tool and selecting the *Attributes* command in the *Drawing Objects* menu. The attributes can also be edited by double-clicking on an object.

#### **Text Attributes**

If a text objet is selected, the *Attributes* command in the *Drawing Objects* menu brings up the dialog shown in Figure 6.14. The dialog can be used to change the font, the color, or the text string itself. An option is also provided to fill a rectangle just containing the text with a user-specified color. This option can be useful to help the text stand out from the objects being drawn behind the text.

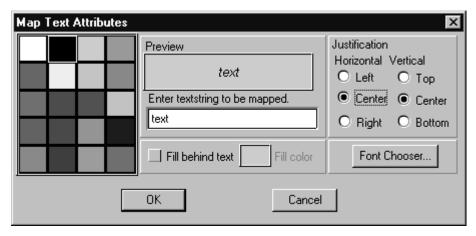


Figure 6.14 The Text Attributes Dialog.

### **Rectangle and Oval Attributes**

The attributes for both rectangles and ovals can be edited with the *Rectangle/Oval Attributes* dialog shown in Figure 6.15. Rectangle and oval attributes include line style, line color, and line width. An option can also be set to either draw only the outline of the rectangle or oval (no fill) or fill the object with a user-specified fill pattern and color.

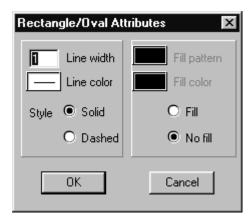


Figure 6.15 The Rectangle and Oval Attributes Dialog.

#### **Line Attributes**

The attributes for lines are edited with the *Line Attributes* dialog shown in Figure 6.16. The line attributes include line color, line width, and line style. The arrowheads associated with a line can also be edited. The length and width of the arrowhead can be defined along with the placement of the arrowheads. The arrowheads can be placed at the beginning of the line, the end of the line, or at both ends of the line.

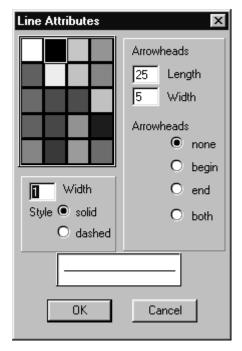


Figure 6.16 The Line Attributes Dialog.

#### **Default Attributes**

When a new object is created, it inherits the default attributes for that object type. The default attributes are defined by selecting one of the drawing object tools (line, rectangle, oval or text) and selecting the *Attributes* command in the *Drawing Objects* menu.

## 6.3.3 Display Options

The *Display Options* command in the *Drawing Objects* menu brings up the dialog shown in Figure 6.17. The dialog is used to toggle the display of text, lines, rectangles, and ovals. Turning off the toggle for an item disables the display of the item but does not delete the item.



Figure 6.17 The Drawing Object Display Options Dialog.

## 6.3.4 Drawing Order

The order in which drawing objects are displayed becomes important whenever a rectangle or oval is displayed in the color fill mode. The order of drawing objects can be controlled using the *Move to Front*, *Move to Back*, *Shuffle Up*, and *Shuffle Down* commands.

#### Move to Front

The *Move to Front* command causes the selected drawing object to be drawn last. In other words it appears on top or in front of all other drawing objects.

#### Move to Back

The *Move to Back* command causes the selected drawing object to be drawn first. In other words it appears at the bottom or in back of all other drawing objects.

#### Shuffle Up

The *Shuffle Up* command causes the selected drawing object to be displayed one object later than it is currently displayed. This causes it to appear in front of the object which is currently being displayed just ahead of it.

#### **Shuffle Down**

The *Shuffle Down* command causes the selected drawing object to be displayed one object sooner than it is currently displayed. This causes it to appear in back of the object which is currently being displayed just behind it.

## 6.4 Images

An image is a digital image such as a scanned map or aerial photo. A common format for saving such images is the TIFF (Tags Image File Format) format. TIFF images can be imported to *SMS* and displayed in the background to aid in the placement of objects as they are being constructed or simply to enhance a plot.

## 6.4.1 Importing an Image

The first step in using a new digital image for either background display or for texture mapping is to import the image. This is accomplished by selecting the *Import* command in the *File* menu. This will open the *Select Import Format* dialog, where you will choose *TIF/GIF*. This brings up the *File Browser* which is used to select the TIFF file. The selected TIFF file must be in the "packbits" compressed format. If your image is not in this format, you will need to convert it using an image processing program such as *XV* (UNIX) or *Paint Shop Pro* (PC).

After selecting the TIFF file in the *File Browser*, the *Register Image* dialog always appears. This dialog is described in the following section.

## 6.4.2 Registering an Image

Before an image can be used for background display or for texture mapping, the image must be "registered". Registering an image involves identifying three points on the image corresponding to locations with known real world (XY) coordinates. Once these points are identified, they are used by *SMS* to stretch or map the image to the proper location when it is drawn with the other objects in *SMS* in the *Graphics Window*. If an image is not registered properly, any objects which are created using the background image as a guide will have the wrong coordinates.

An image is registered using the dialog shown in Figure 6.18. This dialog is used to register a new image as it is being imported. It appears automatically after the *Import* command is selected and a new TIFF file is chosen. It can also be accessed using the *Register* command to change the registration of a previously imported image.

The *Lat/Lon* calculator buttons can be used to convert a latitude-longitude pair into equivalent UTM coordinates.

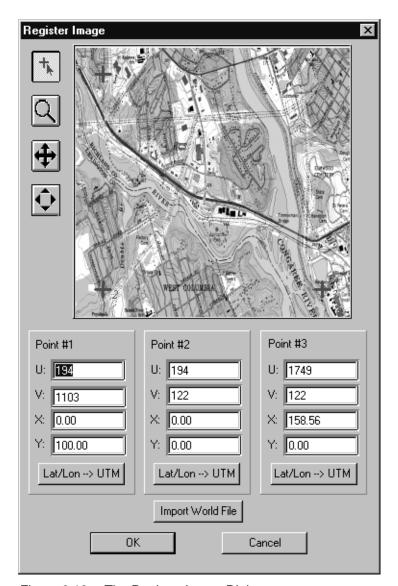


Figure 6.18 The Register Image Dialog.

The main feature of the *Register Image* dialog is a large window in which the image is displayed. Three points (shown by "+" symbols) are also displayed in the window. These points are used to identify locations with known real world coordinates. The real world coordinates (X,Y) and image coordinates (U,V) of the three registration points are listed in edit fields below the image. The points are moved to the desired locations on the image by dragging the points using the tools to the left of the image (described below). Once the points are located, the real world coordinates can be entered in the edit fields shown below the image.

The following tools can be used to help position the registration points:

## Select Point Tool

The *Select Point* tool is used to select and drag register points to a location on the map for which real coordinates are known so that they can be entered in the corresponding XY edit fields.

## **Z**oom Tool

In some cases, it is useful to magnify a portion of the image so that a registration point can be placed with more accuracy. The *Zoom Tool* is used to zoom in a portion of the image.

## Pan Tool

After zooming in on a portion of the image, the *Pan Tool* is used to pan the image vertically or horizontally.

# Frame Macro

The *Frame Macro* is used to automatically center the entire image within the drawing window of the dialog after panning and zooming in on a specific location.

## 6.4.3 Resampling an Image

Once an image is registered, it is positioned and displayed in the *Graphics Window* such that the entire image is visible. As the image is first drawn, the image goes through a process called "sampling". Sampling is a process of converting the image from its actual resolution to the screen resolution. TIFF images typically have a much higher resolution (pixels per inch) than the computer screen. During the sampling process, each pixel on the screen is assigned a color by evaluating the pixels of the TIFF image at the same location. This process can take a few seconds or a few moments, depending on the size of the image and speed of your computer. Once the process is complete, the image is drawn to the screen.

Since the sampling process can take a significant amount of time, it is not convenient to wait for the image to be resampled each time the screen is refreshed or each time the image is panned or zoomed. Thus, to speed up the image display, when the image is first sampled *SMS* stores a bitmap copy of the sampled image at the screen resolution. This bitmap image is then used to refresh the display of the image.

In many cases, the region of interest in an image corresponds to a small sub-region of the image. In such cases, it is possible to focus on the region of interest by zooming in with the *Zoom* tool. However, after zooming in, *SMS* initially displays the image by stretching the bits of the sampled bitmap described above. This results in a grainy image, the graininess depending on the level of magnification. The magnified image can be restored to a high resolution, sharp image by selecting the *Resample* command from the *Image* menu. The *Resample* command repeats the sampling process

described above to generate a screen resolution bitmap image of the currently visible region using the underlying high resolution TIFF image. Of course, since the TIFF image has a limited level of resolution, zooming in too far will result in a situation where the resolution of the screen exceeds the resolution of the TIFF image. In such cases, the *Resample* command is ineffective in increasing the clarity of the image.

After zooming in and resampling an image, it may be necessary to pan the image or zoom back out. When doing so, it will be discovered that the entire image is no longer visible. Only the portion of the TIFF image that was visible when the last *Resample* command was issued is visible. To make a different portion of the image visible, pan or zoom to where the desired region fits in the *Graphics Window* and select the *Resample* command again.

## 6.4.4 Fit Entire Image

As described in the previous paragraph, after zooming in on a small sub-region of the image and selecting the *Resample* command, it may be necessary to zoom back out and resample either the entire image or a different sub-region of the image. The *Fit Entire Image* command is provided to assist this process. When the *Fit Entire Image* command is selected, the visible region is zoomed out so that the entire TIFF image just fits within the *Graphics Window* and the boundary of the TIFF image is shown in red. At this point, the entire image can be resampled or a new sub-region can be zoomed and resampled.

## 6.4.5 Deleting Images

The *Delete* command in the *Images* menu is used to delete the current image.

## 6.4.6 Exporting the Resampled Region

Since TIFF images often have extremely high resolutions, they can require significant memory. For example, a digitized version of a USGS quad sheet can require as much as 40 MB in uncompressed form and 6 MB in compressed form. Fortunately, *SMS* works directly with TIFF images in the compressed form so there is no need to uncompress the entire image. However, memory may still be a concern. Even though *SMS* always works with a sampled bitmap which has a resolution no greater than required by the screen, the entire compressed image must be loaded into RAM whenever the image is resampled.

In many cases, only a portion or sub-region of a large image is needed for a modeling study. If so, the memory requirements can be reduced and the importing and resampling speed can be increased by clipping out the region of the TIFF required for the study prior to importing the image to *SMS*. This can be accomplished by reading the image into a graphics program such as *XV* or *Paint Shop Pro*, clipping out the

region of interest, and saving the clipped region to a separate file. This clipped region can then be imported to *SMS*.

Another option for saving a TIFF image is to use the *Export* command in the *File* menu. This command allows you to save the original TIFF image corresponding to the currently resampled TIFF file. When the *Export* command is selected, the dialog shown in Figure 6.19 appears.

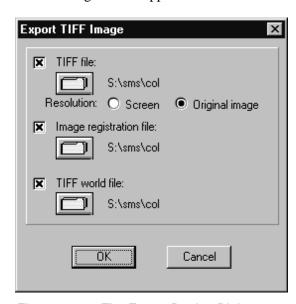


Figure 6.19 The Export Region Dialog.

The TIFF file item is used to designate the name of the new TIFF file that will be created containing the current resampled region. The *Image registration file* option is used to designate the name of the image file corresponding to the resampled image. The image file will include the name of the new TIFF file.

Two options are available for determining the resolution of the new TIFF file. If the *Screen* option is chosen, the TIFF image will be saved at a resolution which matches the screen resolution. If the original TIFF image has a high resolution, this can significantly reduce the memory required to store the resampled region. However, if this option is chosen, once the image is read back in, you will not be able to zoom in and resample the image. If the *Original image* option is chosen, the resampled region is saved using the pixel density of the original TIFF image. This allows you to zoom in on the image and use the *Resample* command after the image is read back into memory.

The last option is to save a *TIFF world file*. This file stores information on the register points. In the register image dialog this file can be imported to automatically assign values to the register points of that image.

## 6.4.7 Export TIFF vs. Save Image

There are two options for saving TIFF images or files related to TIFF images. A summary of the two options is provided here to help reduce confusion concerning the differences between the two commands.

#### **Export TIFF**

When the *Export* command is selected from the *File* menu, one of the options listed in the resulting *Export* dialog is a TIFF file. This option is used to save a TIFF image of whatever is currently being displayed in the *Graphics Window*.

### Save Image

The *Save* command in the *File* menu can be used to save an image file. An image file contains the name of the file containing the TIFF image, the registration points, and the bounds of the currently resampled region. Once the file is saved, an image can be restored by reading the image file using the *Open* command in the *File* menu. *SMS* reads the image file, opens the TIFF file, and registers and resamples the image. This makes it possible to restore an image without having to repeat the registration and resampling process.

## 6.5 DXF Files

In many modeling studies, drawings of the site being modeled are generated in a CAD package such as *AutoCAD*. These drawings can be exported from the CAD package in the DXF format. DXF stands for "Drawing Exchange Format" and is supported by most CAD programs. DXF files can be imported to *SMS* and displayed in the *Graphics Window* to assist in model placement or simply to enhance the display of a model.

## 6.5.1 Importing DXF Files

DXF files are imported to *SMS* using the *Import* command in the *DXF* menu. Once a file is read, the objects in the file are displayed in the *Graphics Window*.

SMS currently supports the R12 (Release 12) version of DXF files. The latest release of AutoCAD is Release 13. DXF files generated by AutoCAD R13 cannot be imported directly to SMS. Before they can be imported, they must be converted to R12 files using the DXFIX utility. DXFIX is a small DOS utility provided with all copies of AutoCAD R13. Consult your AutoCAD documentation for more details.

The R13 DXF format will be supported in future versions of *SMS*.

## 6.5.2 Display Options

The objects in a DXF file are organized into layers. The display of layers in a DXF drawing is controlled using the *Display Options* command in the *DXF* menu. This command brings up the dialog shown in Figure 6.20.

The *DXF Display Options* dialog is different than the display option dialogs of most other entities. The color and style of DXF objects are determined from the application which originally created the drawing file, and cannot be edited within *SMS*. The only option available for DXF objects is the ability to hide or display individual layers. The names of the layers in the drawing are shown in the box on the left of the dialog. An asterisk (\*) appears to the left of the names of the visible layers. The visibility of a layer is toggled on or off by selecting the layer name with the cursor. All of the layers can be made visible with the *Select All* button. Likewise, all of the layers can be hidden with the *Unselect All* button. The *Delete All* button deletes all of the layers.

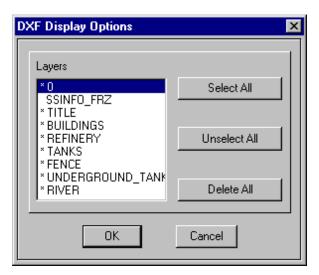


Figure 6.20 The DXF Display Options Dialog.

### 6.5.3 DXF -> Feature Objects

A set of DXF objects which have been imported to *SMS* can be converted to feature objects by selecting the *DXF* -> *Feature Objects* command in the *DXF* menu. DXF points are turned into points, DXF lines and polylines are turned into arcs, and DXF polygons are turned into polygons. The feature objects are added to the active coverage. Once converted, the feature objects can used to build conceptual models.



Figure 6.21 DXF to Feature Objects Conversion Dialog.

### 6.5.4 DXF -> Scatter Points

This function is not yet supported by SMS.

## 6.5.5 Deleting DXF Files

The *Delete* command is used to delete all DXF objects. The DXF objects can also be deleted from within the *DXF Display Options* dialog.

## 6.6 Reading and Saving Map Files

Most of the objects created in the *Map* module can be saved to a map file. Map files are saved by selecting the *Save* command in the *File* menu and selecting the *Map file* option. Feature objects, drawing objects, and the image file are saved in the map file. A map file can be read back into *SMS* using the *Open* command in the *File* menu.

6-36

CHAPTER 7

# River Module **E**

The *River Module* is used to construct 1D river profiles. Tools are provided in this module for creating a "tree" of data to describe the river being modeled. An example of a river tree is shown in Figure 7.1. Currently, the river model includes model specific interfaces for only the *WSPRO* model. Future models may include. As a river section is created, properties can be assigned to it to represent different bed conditions, geometric characteristics or flow parameters. This chapter explains the procedures of creating and editing the river tree, while Chapter 12 explains the *WSPRO* interface.

To open the river window if it is not open, choose *Show River Window* from the *Display* menu. It is also advisable to open the *Plot Window* when creating and editing the river tree by selecting *Show Plot Window* from the *Display* menu. *SMS* can be started with the *River Window* and the *Plot Window* already up by using the command line option *-dm river*. See section 1.2 for more information on using command line arguments.



Figure 7.1 An example of the river window.

#### 

A station represents a position on the river. It has a specific reference distance associated with it.

The *Create Station* tool is used to create a new river station. Using the mouse, click at a desired location in the river tree. (The first station can be created by clicking anywhere within the *River Window*.) The river station icon appears at the position of the new station. When a new station is created, a cross section is also created and its

section icon branches off the station. You will be prompted to enter a section reference distance (SRD) for the new station. The default SRD is halfway between the bounding sections, if the section is interpolated. The Section Editor dialog then appears to allow you to edit the new cross section.

When a new station is created between two existing stations, the cross section geometry is linearly interpolated from the two existing cross sections depending on the SRD you specify. In the same manner, the cross section at a station created before or after all existing stations is copied and extrapolated from that of the first or last station.

# 7.2 Select Station

The *Select Station* tool is used to select any existing station. A station may be selected either by clicking directly on a river station icon, or by dragging a box around it. Multiple selections may be made while holding the SHIFT key. A station is deselected by clicking its icon a second time while holding the SHIFT key or selecting another section without the SHIFT key. A plot of each section of the selected stations will be displayed in the *Plot Window* if it is open. To delete selected stations, choose *Delete* from the *Edit* menu or click the Delete macro.

## 7.3 Create Section Tools

# 7.3.1 Create Bridge Section

The *Create Bridge Section* tool is used to add a bridge section to a station. To add a bridge section, select this tool and click in the river window at the desired station.

The bridge section icon will branch off the station and its section can be edited. Each bridge must have an approach section and an exit section without bridges. This means that bridges cannot be attached to the first or last stations of the river tree. After the bridge is created, the *Section Editor* appears to enable you to edit its parameters.

# 7.3.2 Create Road Section

The *Create Road Section* tool is used to add a road section to a station. To add a road section, select this tool and click in the river window at the desired station. The road

section icon will branch off the station and its section can be edited. Only one road section can be created at each station. After the road is created, the *Section Editor* appears to enable you to edit its parameters.

# 7.3.3 Create Culvert Section

The *Create Culvert Section* tool is used to add a culvert section to a station. To add a culvert section, select this tool and click in the river window at the desired station.

The culvert section icon will branch off the station and its section can be edited. As with the other sections, the *Section Editor* appears to enable you to edit its parameters.

# 7.4 Select Section

A section represents a specific feature at a station.

The *Select Section* tool is used to select any existing section. A section may be selected the same way as a station (see section 7.2). The types of sections that can be selected are cross sections, bridge sections guidebank sections, and culvert sections. These icons appear in the *River Window*. Just as with stations, the selected sections can be deleted by choosing *Delete* from the *Edit* menu or by clicking the Delete macro. Deleting a cross section will delete the entire station at that cross section. Deleting a bridge section will also delete a guidebank section if one is attached to it.

CHAPTER 8

# RMA2 Interface

RMA2 is a hydrodynamic modeling code that supports subcritical flow analysis, including wetting and drying and marsh porosity models. It is part of the TABS analysis package supported by the U.S Army Corps of Engineers Waterways Experiment Station (USACE-WES). The methods of analysis used by the TABS codes along with their file formats and input parameters are described in their own documents. SMS supports both pre- and post-processing for RMA2. RMA4 is a constituent migration model that utilizes the flow field generated by RMA2. SMS only supports post-processing for RMA4.

A mesh for use with *RMA2* is generated and edited using the *Mesh Module*. *RMA2* specific modeling parameters are generated and applied to the mesh using the commands that are grouped in the *RMA2* menu. Those commands are described in this chapter. Post-processing is generic, once the solution file for *RMA2* has been imported into the Data Browser.

## 8.1 Open Geometry

The *Open Geometry* command in the *RMA2* menu reads in a geometry file that has been previously created and saved. Geometry files typically have the file extension ".geo". The name of the current geometry file is displayed at the top of the *Main Graphics Window*. The geometry file stores the location of nodes that define the corners of the elements, and the connectivity of nodes to make the elements for a mesh in *RMA2* format. The nodes that define the midsides of the elements are only required if the element has curved sides. See the TABS Primer for an example of the geometry file format. Use the file selection dialog to choose an existing file. Opening a new geometry file, as with selecting *New* from the *File* menu causes all existing mesh data (geometry and boundary condition data) to be deleted from memory.

# 8.2 Save Geometry

The *Save Geometry* command in the *RMA2* menu saves a geometry file so that it can be opened at a later time or used in an analysis.

# 8.3 Open BC

The boundary condition data of an *RMA2* numerical model is stored in a boundary condition file. This data includes the definition of inflow and outflow boundaries of the mesh, run control parameters such as time steps and various other model specific information. Boundary condition files typically have the file extension ".bc". The *Open BC* command in the *RMA2* menu reads in a boundary condition file that has been previously created and saved. Opening a boundary condition file causes all existing boundary conditions to be deleted. See Chapter 3 of the TABS Primer for an example of the boundary condition file format.

## 8.4 Save BC

The Save BC command in the RMA2 menu saves a boundary condition file so that it can be opened at a later time or used in an analysis.

# 8.5 Assign Nodal BC

The *Nodal Boundary Conditions* dialog (see Figure 8.1) may be used to assign boundary conditions to individual nodes. Before assigning boundary conditions to nodes, at least one boundary node must be selected using the *Select Nodes* tool. All

selected nodes must be on the mesh boundary. Attempting to assign nodal boundary conditions to an interior node will result in an error message.

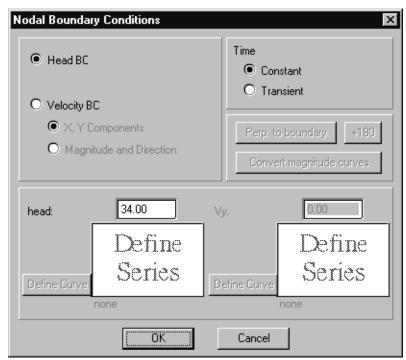


Figure 8.1 Assign Nodal Boundary Conditions Dialog.

All selected nodes are assigned identical conditions. If different boundary conditions are desired, the user must either select one node at a time and then assign its conditions, or if a range of conditions are desired, the nodal boundary conditions can be interpolated using the *Interpolate Nodal BC* command in the *Node* menu.

To assign constant boundary conditions to the selected node(s), select the *Constant* radio button in the *Time* section of the *Nodal Boundary Conditions* dialog and enter the value into the available edit field(s). To assign transient (time-dependent) boundary conditions to the selected node(s), select the *Transient* radio button and click the *Define Curve* button. For dynamic head boundary conditions, only the left curve is active. For dynamic velocity boundary conditions, both curves are active. The velocity can be entered as a combination of magnitude and direction, or in terms of its *x* and *y* components. Clicking on a *Define Curve* button will bring up the *XY Series Editor* dialog described in Chapter 13. This editor allows the user to define a function of values -vs.- time.

## 8.5.1 Head Boundary Conditions

To assign head boundary conditions to the selected node(s), select the *Head BC* radio button and enter the appropriate value(s) or define the curve(s).

#### 8.5.2 Velocity Boundary Conditions

To assign velocity boundary conditions to the selected node(s), select the *Velocity BC* radio button. Velocity boundary conditions may be defined as either vector components in the x and y directions, or as a magnitude and direction. If using the magnitude and direction method, selecting the *Perp. to boundary* button will calculate an angle which is perpendicular to the boundary at the selected node(s). Clicking the +180 button will add 180 degrees to the steady-state angle(s). The angle is measured in degrees from the positive x-axis to the direction in which the vector points. Since velocity boundary conditions are saved as vector components in the x and y directions, any steady-state velocity boundary conditions defined as a magnitude and direction are automatically converted into vector components. To convert dynamic magnitude and direction curves into the vector component curves, click the *Convert magnitude curves* button.

# 8.6 Assign String BC

The *Nodestring Boundary Conditions* dialog (see Figure 8.2) may be used to assign boundary conditions to nodestrings. Before assigning boundary conditions to strings, at least one nodestring must be selected using the *Select Nodestrings* tool. All selected strings must be on the mesh boundary. Attempting to assign a boundary condition to an interior nodestring will result in an error message.

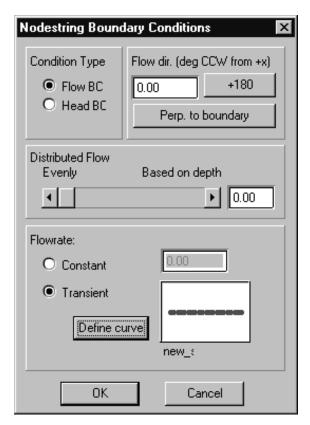


Figure 8.2 Nodestring Boundary Conditions Dialog.

To assign constant boundary conditions to the selected nodestring(s), select the *Constant* radio button in the *Time* area of the *Nodestring Boundary Conditions* dialog and enter the value into the available edit field(s). To assign transient (time-dependent) boundary conditions to the selected nodestring(s), select the *Transient* radio button and click the *Define Curve* button. This will bring up the *XY Series Editor* described in Chapter 13. This dialog allows the user to define a function of values -vs.- time.

## 8.6.1 Flow Boundary Conditions

To assign flow boundary conditions to the selected nodestring(s), select the *Flow BC* radio button and enter the appropriate values or define the curve. In addition to the flow rate, the user must specify a flow direction and a distribution factor. The direction is the angle in degrees measured from the positive x-axis to which the water is flowing. Clicking the *Perp. to boundary* button will calculate an angle which is perpendicular to the boundary at the approximate center of the most recently selected nodestring. Clicking the +180 button will add 180 degrees to this angle. The distribution factor allows the flow to be distributed by depth rather than equally along the length of the nodestring. The user specifies a value using a scroll bar. The ends of the scroll bar correspond to the two extremes, length distribution and depth distribution. Values in the middle blend the two methods.

#### 8.6.2 Head Boundary Conditions

To assign head boundary conditions to the selected nodestring(s), select the *Head BC* radio button and enter the appropriate value or define the curve.

### 8.7 Delete BC

The *Delete BC* command will delete any boundary conditions previously assigned to any selected nodes or any selected nodestrings. If neither nodes nor nodestrings are selected, you will be given the option to delete the boundary conditions from all the nodes, all the nodestrings, or both. If a boundary condition is deleted from a nodestring, the nodestring will not be deleted, but will be defined as a generic nodestring.

# 8.8 Add GC String

Any selected nodestring(s) can be defined as a geometry continuity (GC) string by selecting the string and invoking the *Add GC String* command from the *RMA2* menu. GC strings are used by TABS-MD software to compute the flow rate through a sequence of edges defined by the strings.

For best results, GC strings should be input perpendicular to the anticipated flow direction and extend from one side of the mesh to the other. SMS creates a GC string automatically for each boundary condition string. GC strings and boundary strings are written out to the boundary condition file.

### 8.9 RMA2 Materials

The *RMA2 materials* menu item allows you to create new materials or edit existing materials and to assign *RMA2* specific material properties to materials (see Figure 8.3).

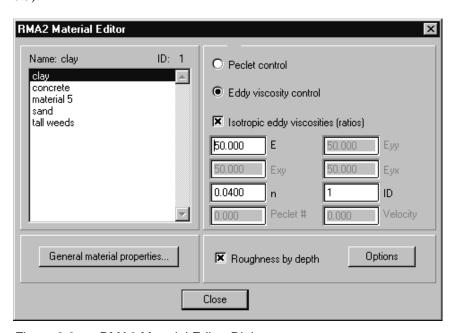


Figure 8.3 RMA2 Material Editor Dialog

Materials are assigned to elements in the *Elements* menu. The *RMA2* properties associated with each material are the Manning's n coefficient and either the eddy viscosity or Peclet number. If the user is specifying eddy viscosities, four values may be entered, corresponding to:

- x-momentum turbulent exchange in the x direction (Exx).
- x-momentum turbulent exchange in the y direction (Exy).
- y-momentum turbulent exchange in the x direction (Eyx).
- y-momentum turbulent exchange in the y direction (Eyy).

Similar values are available for specifying ratios for automatic eddy viscosity computation based on Peclet number. Since these values apply to all elements which reference this material, it is recommended that the user utilize the *Isotropic eddy viscosities (ratios)* option. This option forces a uniform eddy viscosity in all directions. If all elements of a particular material type are aligned to the principal axes, directional viscosities can be applied.

Clicking the *General Material Properties* button will bring up the *Materials Editor* dialog (see Section 2.8.4). In this dialog, you can specify other options of the selected

material, such as the color and pattern of the material or the name of the material. Materials are also created and deleted using this dialog.

#### 8.10 Model Check

A *Model Check* should be performed on all *RMA2* models before attempting an analysis. The model check will perform a basic check to insure that all of the needed information to run the analysis is present. The *Model Check* command in the *RMA2* menu causes the *Model Check* dialog (see Figure 8.4) to appear.

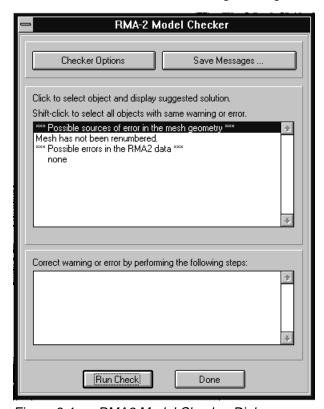


Figure 8.4 RMA2 Model Checker Dialog

Selecting the *Checker Options* button will cause the *RMA2 Model Checking Options* dialog to appear. This dialog lists the checks that may be performed during the m model checking procedure. By default all supported checks are enabled. The checks include:

- Check Water Surface Elevation. The initial water surface elevation is considered invalid if it is higher than the highest node and the dry element flag has not been set (DE card, see section 8.12.3).
- Check Boundary Conditions. Boundary conditions should consist of at least a head boundary condition (assigned to a node or a string). Flow and velocity boundary conditions can also be defined (see sections 8.5 and 8.6).

Mesh Check Options. Generic model checking options include: renumbered
mesh, material definition, duplicate nodes and small voids or small
unexpected interior mesh boundaries. In this dialog you can also choose to
show all or limit the number of similar messages to report.

Click the *Run Check* button to perform a model check with the currently selected options. After *SMS* has completed its model check, the error messages will appear in the top text window. Some of these error messages will include directions on how to fix the error. To read these directions in the lower text window, click on the error message. To save this information to a text log file, click the *Save Messages* button and choose a file to save the information in. To close the *Model Checker*, click the *Done* button.

## 8.11 Global BC Control

The *RMA2* numerical model requires several user specified parameters to control the analysis. These include such things as the control of files used for input and output and the control of time simulation for dynamic analysis. The user controls these parameters using the *Global BC Control* dialog accessed by the *RMA2 Control* command in the *RMA2* menu (see Figure 8.5). The *RMA2* analysis model supports around 50 different types of cards to control the definition and analysis of a model. A majority of these card types are very specialized and rarely used. For this reason, *SMS* does not support them. If a user wants to use one of these specialized card types, it must be added to the boundary condition file after *SMS* has finished its preprocessing operation. As the boundary condition file is read in, any cards that are not supported by *SMS* are saved internally, connected to their time step. These cards are then written to the boundary condition file in the appropriate time step when *SMS* outputs a boundary condition file. A complete list of all cards available for geometry files is contained in the TABS Primer.

Each field in the dialogs associated with *RMA2* are connected to an card type. The dialogs in *SMS* provide the user with a textual description of what the variables are used for in *RMA2*. The exact card for a specific field is displayed in *SMS* using the interactive help messages. As the cursor moves over that field, the card is displayed along with a message in the help portion of the *Edit* dialog.

8-10

Figure 8.5 Global BC Control Dialog.

Input parameters for the commonly required cards are specified in the *Global BC Control* dialog (see Figure 8.5). Default values are provided for all required cards.

### **8.11.1 Job Title**

The *Job Title* section of the *Global BC Control* dialog allows the user to give a title to and make comments about the problem being modeled. These fields correspond to the T1, T2, and T3.

#### 8.11.2 File Control

The *Files* section of the *Global BC Control* dialog allows the user to controls what file I/O will occur during the analysis. This group of toggle boxes lets the user specify whether *RMA2* will read or write hot start files, use alternate boundary conditions, and output runtime messages to a print echo file. The changes that must be made for dynamic analysis are discussed in Chapter 3 of the TABS Primer. *SMS* 

actually assigns logical unit numbers for each file that is toggled on. These unit numbers are stored in the \$L card.

#### 8.11.3 Iteration Control

The *Iterations* section of the *Global BC Control* dialog allows the user to control the number of iterations *RMA2* uses at each step of analysis for both steady-state and dynamic analysis. This corresponds to the TI card in the boundary condition file. If the default values assigned by *SMS* do not provide sufficient stability, the iteration number may be increased. The user also has control of a depth convergence criteria for both steady-state and dynamic analysis.

#### 8.11.4 Computation Time Control

The *Computation time* section of the *Global BC Control* dialog allows the user to control the simulation time. This corresponds to the TZ card which defines the computation time lengths, the number of time-steps, and the time-step lengths for a dynamic analysis. The user also can specify for an analysis to pick up at a time step other than the starting time and use a hotstart file. The type of analysis (steady-state - vs.- dynamic) can be set in the *Solution type* section.

#### 8.11.5 Units Control

The *Units* section of the *Global BC Control* dialog allows the user to control the type of units used during the analysis. This corresponds to the SI card.

### 8.11.6 Other Options Control

The *Other Options* section of the *Global BC Control* dialog allows the user to control the fluid temperature (FT card) and gravity constant. It also includes the *Optional BC* control button which causes the *Options BC Controls* dialog to appear. The fluid temperature is always specified in degrees Celsius.

# 8.12 Optional BC Control

The *Optional BC Control* dialog, allows the user to set the most commonly used optional global boundary conditions. Since these features are optional, toggle boxes appear in the upper right corner of each section. If the toggle is not selected, the card is not written to the boundary condition file, and the default value is used. The options include machine type, geometric editing, element wetting and drying, print tracing, and the specification of a global Peclet number (see Figure 8.6).

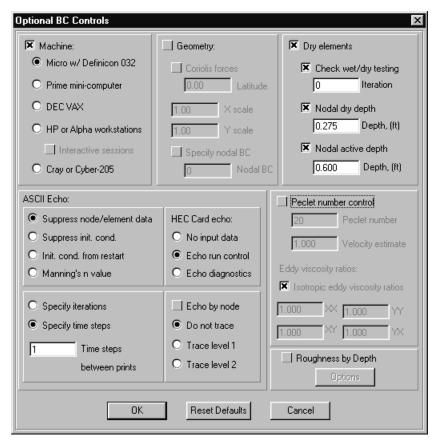


Figure 8.6 Optional Boundary Condition Control Dialog.

## 8.12.1 Machine Type

The *Machine* section of the *Optional BC Controls* dialog allows the user to specify what machine type will be used to perform the numerical analysis. In *RMA2*, this corresponds to the \$M card. If this toggle is not selected, Alpha/HP workstation is the default value.

## 8.12.2 Geometry Modifications

The *Geometry* section of the *Optional BC Controls* dialog allows the user to specify global geometric modifications to the geometry. These modifiers make up the BCC, GS and LA cards. Specifically the user can instruct *RMA2* to include the Coriolis forces effect of the earth's motion, scale the geometry in either Euclidean direction, and limit the number of nodal boundary conditions applied.

#### 8.12.3 Wetting and Drying

The *Dry Elements* section of the *Optional BC Controls* dialog allows the user to specify how *RMA2* will handle wetting and drying elements during the analysis. This corresponds to the DE card. This card controls the defining criteria for wetting and drying of elements. In addition, the card controls how often *RMA2* checks elements to determine their wet or dry status.

*RMA2* also supports the marsh porosity method of handling the wetting and drying problem. The DM card must be added to the bc file by hand after SMS writes the bc file. The card is preserved from run to run by SMS.

Since no interface for marsh porosity exist in *SMS*, details for the option can be found in the *RMA2* manual.

#### 8.12.4 Peclet Number Control

*RMA2* allows the user to control the eddy viscosity of the mesh or regions of the mesh using materials, or it will compute the values automatically from a Peclet number and the velocity. If the *Peclet number control* is used (PE card), the user can specify a Peclet number for the entire mesh. If a PET card is desired to specify a Peclet number for a material, the user should specify this in the *RMA2 Materials* dialog.

#### 8.12.5 Echo Control

The ASCII echo region controls the output (TR card) generated during an *RMA2* analysis. The user can specify what suppression type is desired, what HEC card echoing is desired, how frequently a print should be made, and whether nodes should be traced.

# 8.13 Display Options

Selecting the *Display Options* command from the *RMA2* menu allows the user to specify the display options to items that are specific to the *RMA2* analysis model. The display color of an item is changed by clicking the color box next to that item. The items that can be specified in the *RMA2 Display Options* dialog are:

- The *Nodal BC Symbol* option will toggle the display of the symbol placed at all nodes where nodal boundary conditions have been defined. Clicking the color box will allow you to change the symbol used.
- The *Nodal velocity BC vectors* option will toggle the display of the vectors placed at nodes where velocity boundary conditions have been assigned.

- The Nodestring head BC symbol option will toggle the display of the head symbol placed near the center of nodestrings where head boundary conditions have been defined.
- The *Nodestring flow BC symbol* option will toggle the display of the flow symbol placed near the center of nodestrings where flow boundary conditions have been defined.

Clicking the *General Display Options* button will bring up the *Mesh Module Display Options* dialog where the display of nodes, elements, contours, and other items are controlled.

CHAPTER 9

# SED2D-WES Interface

SED2D-WES is the name for the sediment transport numerical model developed and supported by the U.S. Army Corp of Engineers Waterways Experiment Station. It had been distributed previously under the name of STUDH. SED2D-WES has the ability to compute sediment loading and bed elevation changes when supplied with a hydrodynamic solution computed by RMA2. The model supports both clay and sand beds individually, but the two bed types cannot be contained within the same model.

Inside of *SMS*, the *SED2D-WES* menu is disabled until an *RMA2* mesh and boundary conditions file have either been read or saved. This is due to the fact that *SED2D-WES* relies on the existence of an *RMA2* solution, and because some of the control variables defined for *RMA2* are applied to the *SED2D-WES* model. The user can define *SED2D-WES* parameters as soon as the mesh exists. However, it is recommended that *SED2D-WES* pre-processing be done after the *RMA2* analysis has been completed. All *SED2D-WES* solutions are computed in metric units.

#### 9.1 SED2D-WES File I/O

Sediment transport data describing such parameters as the deposition rate and the sediment load, along with the bed conditions and layer definitions used by SED2D-WES are stored in an ASCII file. This file is referred to as a sediment transport file and has a .sed file extension. SED2D-WES is run in an iterative fashion. The original file for a problem will be created by SMS. As SED2D-WES computes the scour and deposition of sediment across the mesh, a new bed definition is created for the problem, and SED2D-WES outputs a new sediment transport data file for successive runs.

#### 9.1.1 New Simulation

The *New* command deletes all *SED2D-WES* data currently in *SMS*. This allows the user to start over in the definition of sediment transport parameters and bed definition and characteristics. This command does not delete the geometry or RMA2 boundary conditions.

## 9.1.2 Open Simulation

The *Open* command allows the user to read in an existing *SED2D-WES* sediment transport file to apply to the geometry.

#### 9.1.3 Save Simulation

The Save command allows the user to save a sediment transport file for SED2D-WES.

### 9.2 Global Parameters

To initialize the definition of a sediment transport problem, global parameter values need to be specified. Any region of the model that is not redefined with local parameters inherits these global parameters. These global parameters include: bed type, diffusion coefficients, initial concentration and settling velocity (see Figure 9.1).

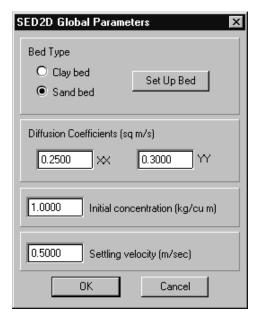


Figure 9.1 SED2D-WES Global Parameters Dialog.

#### **9.2.1 Bed Type**

SED2D-WES supports problems involving either clay beds or sand beds. Natural conditions may involve both, but each must be modeled separately. SMS allows the user to select which bed type is desired. Based on the selected type, the Set Up Bed button invokes the appropriate bed parameter dialog and allows the user to define the default bed condition parameters for the model.

#### Clay Bed

Clay beds are defined by three general parameters, and consist of up to ten layers of clay. The three parameters which define a clay bed are:

- critical shear stress for deposition.
- critical shear stress for particle erosion.
- constant for the erosion equation.

The user must also define the maximum number of layers defined at any location in the model in the *Global Clay Layers* dialog (see Figure 9.2). For each layer, the following parameters are specified:

- thickness
- particle erosion shear stress
- particle erosion constant
- layer erosion shear stress
- one year old layer erosion shear stress
- initial dry density
- one year old dry density
- consolidation coefficient
- layer age

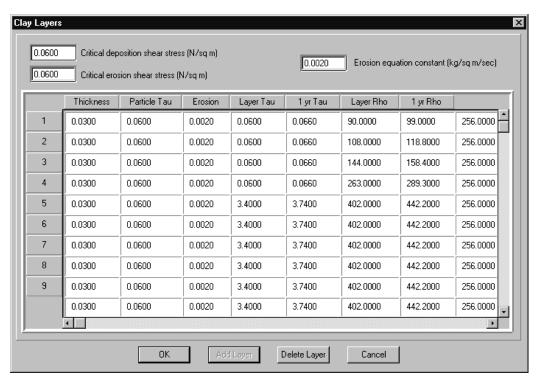


Figure 9.2 Clay Layers Dialog.

#### Sand Bed

Sand beds are defined by nine parameters, specified through the *Global Sand Layers* dialog (see Figure 9.3). These include:

- minimum and maximum grain size.
- specific gravity of the sand.
- grain shape factor.
- characteristic deposition and erosion length factors.
- thickness of the sand layer.
- effective sand grain size for roughness and transport.

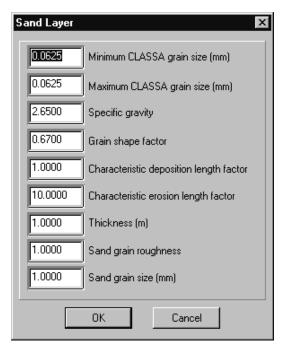


Figure 9.3 Sand Layers Dialog.

#### 9.2.2 Diffusion Coefficients

The *Diffusion Coefficients* section of the *SED2D-WES Global Parameters* dialog allows the user to specify effective diffusion coefficients to be used by *SED2D-WES*. These values are similar to the eddy viscosity parameters used by *RMA2*, and define the turbulent exchange coefficients in each of the principle directions. It is recommended that the two values be identical.

#### 9.2.3 Initial concentration

The *Initial concentration* section of the *SED2D-WES Global Parameters* dialog allows the user to specify an approximation to the initial sediment load in the model.

## 9.2.4 Settling velocity

The *Settling velocity* section of the *SED2D-WES Global Parameters* dialog allows the user to specify the settling velocity for the sediment transport model.

#### 9.3 Local Parameters

The conditions specified as global are applied across the mesh. SMS allows the user to override global conditions for specific nodes, elements, nodestrings, or material

types using the local parameters dialog. While the user can specify these parameters by any object or group, all the data is converted to nodal information inside *SMS*, and the output files specify all parameters either globally or by node. This also applies to parameters that are read in from sediment transport files that may specify local parameters by element or material.

As with the global parameters, the local parameters dialog varies based on the bed type (see Figure 9.4 and Figure 9.5). The user simply selects the regions for which the parameters are specified by selecting the nodes, nodestrings, elements, or material type, and then specifies the value for the desired parameter(s). Unspecified parameters continue to refer to the global value. The user may specify the diffusion constants, initial concentrations, and fall velocity for both clay and sand models. In addition, for clay models the user can specify local layer thickness and age for any layer defined in the global parameters (see Figure 9.4). For sand models the grain roughness and size can be modified for regions, as well as the layer thickness (see Figure 9.5).

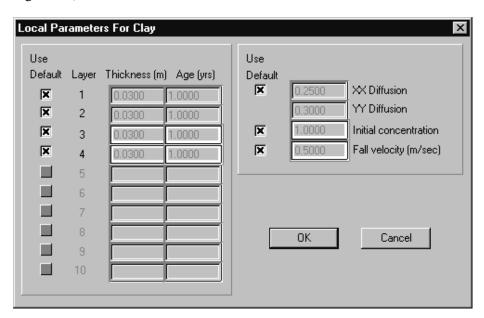


Figure 9.4 Local Parameters for Clay Layers Dialog.

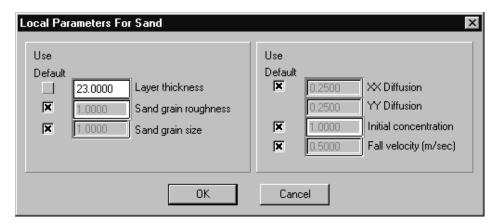


Figure 9.5 Local Parameters for Sand Layer Dialog.

### 9.4 BC Concentrations

SED2D-WES requires sediment concentration to be specified at all inflow boundary nodes. This is accomplished by selecting the boundary node(s) or nodestring and then selecting the BC Concentrations command from the SED2D-WES menu. This concentration can be specified as either constant or transient. A transient value is defined using the Dynamic Boundary Conditions editor (see Chapter 13).

### 9.5 Model Control

SED2D-WES requires several additional parameters to control how the numerical model functions. These parameters are defined using the *Model Control* dialog (see Figure 9.6). The SED2D-WES Model Control dialog provides control of four different parameters. These parameters are:

- Hydraulic Bed Shear Stress
- Crank Nicholson Theta
- Time Control
- Test Convergence

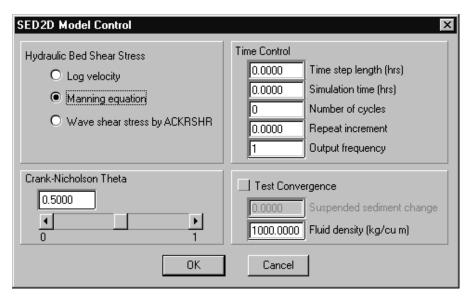


Figure 9.6 The SED2D-WES Model Control Dialog.

### 9.6 Print Control

The *Print Echo Control* dialog (Figure 9.7) allows the user to specify what information is echoed to an output file and how often that information is output. The user also may specify a subroutine trace which outputs the name of the analysis code subroutines as they are encountered. This tool is useful if model instabilities occur and modifications must be made.

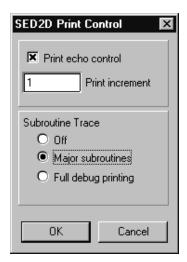


Figure 9.7 The SED2D-WES Print Control Dialog

## 9.7 SED2D-WES Display Options

Specific *SED2D-WES* display options can be set by selecting the *Display Options* command from the *SED2D-WES* Menu (see Figure 9.8). These include:

- The SED2D-WES parameter functions item is used to tell SMS to create functional data for each parameter used in SED2D-WES analysis. This includes layer thickness and age for each layer of a clay simulation and sand roughness, grain size, and layer thickness for a sand simulation. For either bed type, a data set is created for the dispersion coefficients, the initial concentrations, and the fall velocity across the mesh. Using these data sets, contours can be displayed on the mesh to graphically illustrate bed parameters.
- The *Boundary conditions* item lets the user select the color and symbol used to display where boundary node concentrations are defined.

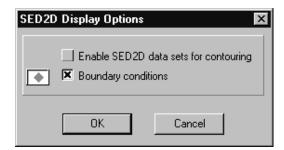


Figure 9.8 The SED2D-WES Display Options Dialog

#### 9.8 Model Checker

The SMS interface to SED2D-WES is equipped with a Model Checker. This feature evaluates the current model, and performs standard checks to assure the quality of the model for numerical analysis. The Model Checker is an ongoing development. Additional checks will be added as suggestions are provided by SMS users, and rules are developed to safeguard against instabilities. The current version of SMS includes an option to assure that concentration values are specified at all inflow boundary nodes and then guide the user to overlooked nodes. The SMS model checker also includes the mesh quality model checking options since SED2D-WES requires a preliminary run by RMA2. Any model checking option may be disabled by the user, but this is not recommended.

CHAPTER 10

# **HIVEL Interface**

HIVEL is a two-dimensional model used to analyze high velocity flow in concrete-lined channels with hydraulically steep slopes. It was developed at the U.S. Army Corps of Engineers Waterways Experiment Station (USACE-WES) to deal specifically with both supercritical and subcritical flow fields, as well as the transition between them. HIVEL is supported and maintained by WES.

This chapter describes the commands used in SMS to create and edit the *HIVEL* specific parameters included in the *HIVEL* menu. For information on using SMS to create the finite element mesh, see chapter 4. Once the analysis is complete, post-processing of the analysis results is performed by importing the solution file using the data browser. (See Lesson 9 of the *SMS Tutorials* and the *HIVEL2D User Manual* for more about running *HIVEL*).

## 10.1 New Simulation

The *New Simulation* command in the *HIVEL* menu deletes all of the *HIVEL* specific data associated with the current model. This data includes the finite element mesh and boundary conditions, model control data, hot start data, and all solutions that have been imported into SMS. Boundary conditions include inflow/outflow specifications and material properties. Model control data includes simulation titles, computation time, and default constants such as gravity and turbulence coefficients.

## 10.2 Open Simulation

The Open Simulation command in the HIVEL menu reads a "super" file which lists data files that have been previously created and saved. These files typically have the file extension ".sup". The name of the current simulation is displayed at the top of the Graphics Window. The data files contain the finite element mesh, hotstart input data, boundary conditions, and model control data for a HIVEL analysis. (See the HIVEL User Manual for more information of data file formats). Opening a new simulation causes the existing finite element mesh and boundary conditions to be deleted from memory, however, data in other formats (such as map module data) is not affected.

## 10.3 Save Simulation

The Save Simulation command in the HIVEL menu opens up the HIVEL Data Files dialog (See Figure 10.1).

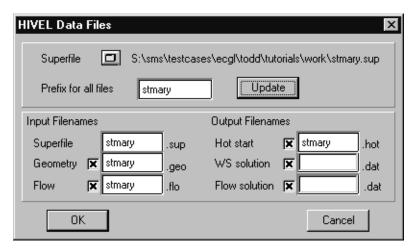


Figure 10.1 HIVEL Data Files Dialog

Specified in this dialog is the superfile along with all optional files that can be created. The path where the files will be saved is shown at the top of the dialog and may be changed by clicking on the file button to the left. After clicking the OK button, the data files can be opened at a later time or used in an analysis. (Note: HIVEL can only be run with linear elements. if there are quadratic elements in the mesh, they will need to be changed to linear before saving a simulation for HIVEL.)

#### 10.4 Build Hot Start

Some models have an option to use a hotstart file. In the HIVEL model, a hotstart file is necessary. The Build Hot Start command in the HIVEL menu invokes the Setup HIVEL Hot Start File dialog (see Figure 10.2).

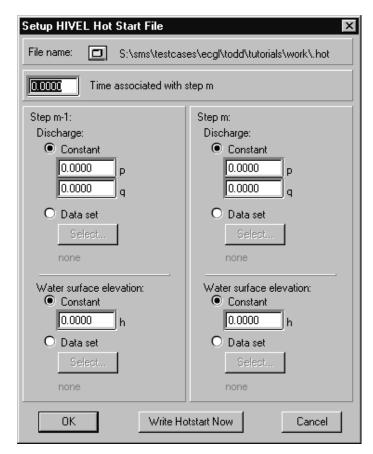


Figure 10.2 Setup HIVEL Hot Start File Dialog

At the top of the dialog, is the name of the file that will used to save the hotstart data. This is the same name that appears in the dialog to save hivel as described above. Under the file name is a field for *Time associated with step m*. This corresponds to the last time step from a previous *HIVEL* run. This should be used as 0.0 for an initial solution.

The rest of the dialog is used to define the last two time steps from the previous model run in order to initialize a new run. For an initial run, these values must be estimated based on experience. It is suggested that an initial velocity be given as zero and the initial water surface elevation be the same as the specified outflow condition.

This initial data can defined as constant for each node or a dataset can be imported and used. When opening a simulation which was previously saved, datasets will be generated from the existing hotstart file. Vector datasets only can be used for velocity in the hotstart file, while scalar datasets only can be used for water surface elevation.

It is important to understand that in addition to being used as inputs to a run, *HIVEL* will overwrite it with the data from the last two time steps of the run. Therefore, if it is desired that the data in a hotstart file be saved, a backup copy should be created.

# 10.5 Assign BC

The *HIVEL Boundary Condition* dialogs (see Figure 10.3 and Figure 10.4) may be used to assign boundary conditions to individual nodes or nodestrings. Before assigning boundary conditions, at least one boundary node or nodestring must be selected. All selected items will receive identical boundary conditions. Attempting to assign boundary conditions to interior nodes or nodestrings attached to interior nodes will result in an error message.

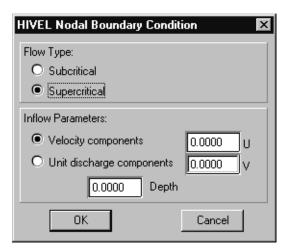


Figure 10.3 HIVEL Nodal Boundary Conditions Dialog

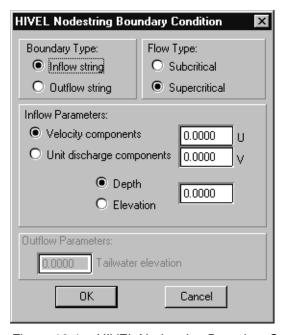


Figure 10.4 HIVEL Nodestring Boundary Condition Dialog

Boundary conditions are assigned as either inflow or outflow and either subcritical or supercritical.

#### 10.5.1 Inflow Boundary

To assign inflow boundary conditions, select the *Inflow* radio button and either *Supercritical* or *Subcritical*. For either of these, enter velocity or unit discharge in the x and y directions. If an inflow boundary condition is specified as supercritical, also enter the water depth or water surface elevation. Inflow can be assigned to either nodes or nodestrings.

## 10.5.2 Outflow Boundary

To assign outflow boundary conditions, select the *Outflow* radio button and either *Supercritical* or *Subcritical*. Tailwater elevation is specified if it is subcritical. Otherwise nothing is specified. Outflow can be assigned to nodestrings only.

### 10.6 Delete BC

The *Delete BC* command will delete any boundary conditions previously assigned to all selected items. If neither nodes nor nodestrings are selected, you will be given an error message that nothing is selected.

## 10.7 Model Control

The *HIVEL* numerical model requires several user specified parameters to control the analysis. These parameters are set and changed using the *HIVEL Model Control* (see Figure 10.5) dialog accessed through the *Model Control* command in the *HIVEL* menu.

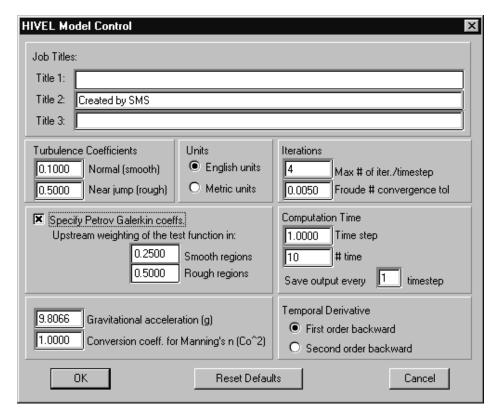


Figure 10.5 HIVEL Model Control Dialog

The rest of this section describes each of the model control inputs and what they are used for.

#### 10.7.1 Job Titles

The *Job Titles* section allows the user to give up to three titles and/or comments about the problem being modeled. These fields correspond to the T1, T2, and T3 cards. Although the *HIVEL* model allows any number of T1 and T2 cards to be created in the boundary condition file, only one of each is stored by SMS.

#### 10.7.2 Turbulence Coefficients

The *Turbulence Coefficients* section allows the user to specify the turbulence coefficient that will be used for computations both near and away from rough flow conditions such as a hydraulic jump. These values, stored in the *turb* record, are used by *HIVEL* to determine turbulent eddy viscosity based on depth, velocity magnitude, and roughness. Default values are 0.10 for smooth flow conditions and 0.50 for rough flow conditions.

#### 10.7.3 Units Control

The *Units* section allows the user to control the type of units used during the analysis. This corresponds to the SI card. When the units of a model are changed, the gravity and manning's n coefficients are automatically converted to the correct new units.

#### 10.7.4 Petrov-Galerkin Coefficients

The *Petrov-Galerkin Coefficients* section allows the user to specify the Petrov-Galerkin weight coefficients for the model. These values are stored in the *pgwc* record. If no values are specified, the defaults of 0.25 and 0.50 for smooth and rough regions, respectively, will be used.

## 10.7.5 Gravity and Manning's Conversion Constants

In the lower left of the dialog, two constants for the model are specified. First, the acceleration of gravity is specified (*grav* record). The initial value for this is given as 32.189 for English units or 9.8066 for Metric units. Second, an empirical conversion coefficient for use in Manning's equation is specified (*mcon* record). The initial value is 2.208 and 1.00 for English and Metric units, respectively. When specifying these, be careful that the units are correctly set (English vs. Metric) because the values will be converted if the units are changed.

#### 10.7.6 Iterations

The *Iterations* section allows the user to specify both the maximum number of iterations that will be performed for each time step and the tolerance of convergence to move on to the next time step (*giterrav* record). Convergence is reached when the maximum change of Froude number at a node from the previous iteration is less than the specified tolerance. If the convergence criteria is not reached at a time step, the solution given may not be valid.

#### 10.7.7 Computation Time

The Computation Time section allows the user to specify the time step in hours (tims record) and the number of time steps HIVEL should perform (step record). The initial time step is that specified in the hotstart file plus one time step from the specification here. Also specified here is the frequency at which solution data will be saved. The default value is 1, meaning the solution will be saved at each time step.

#### 10.7.8 Temporal Derivative

The *Temporal Derivative* section allows the user to specify if first order backward differencing or second order backward differencing will be used to calculate initial guesses at each time step.

#### 10.7.9 Reset Defaults

The *Reset Defaults* button will reset all values to their default values. Customizing the defaults is accomplished by first entering all desired values and then selecting *Save Environment* from the *File* menu. All values that were entered will become the new defaults. To return to factory defaults, it is necessary to delete the SMS.INI file (see section 2.7.8).

# 10.8 Material Properties

The *HIVEL materials* menu item allows you to edit the materials data (Figure 10.6).

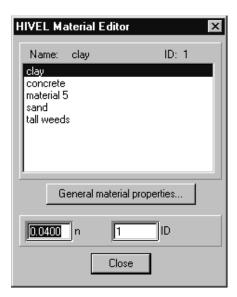


Figure 10.6 HIVEL Material Editor Dialog

Materials are assigned to elements using the *Elements* menu. In *HIVEL*, these materials will have an id value and a value for Manning's n (*mtyp* record ).

Clicking the *General Material Properties* button will bring up the General *Materials Editor* dialog (see Section 2.8.4). In this dialog, you can specify other options of the selected material, such as the color and pattern of the material or the name of the material. Materials are also created and deleted using this dialog.

# 10.9 Display Options

Selecting the *Display Options* command from the *HIVEL* menu allows the user to specify the display options to items that are specific to the *HIVEL* analysis model (see Figure 10.7).

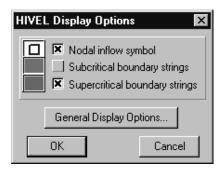


Figure 10.7 HIVEL Display Options Dialog

The display color of an item is changed by clicking the color box next to that item. The items that can be specified in the *HIVEL Display Options* dialog are:

- The *Supercritical inflow* option will toggle the display of the symbol placed at all inflow nodes and nodestrings where supercritical flow was specified.
- The *Subcritical inflow* option will toggle the display of the symbol placed at all inflow nodes and nodestrings where subcritical flow was specified.
- The *Supercritical outflow* option will toggle the display of the symbol placed at all outflow nodes and nodestrings where supercritical flow was specified.
- The *Subcritical outflow* option will toggle the display of the symbol placed at all outflow nodes and nodestrings where subcritical flow was specified.

Clicking the *General Display Options* button will bring up the *Mesh Module Display Options* dialog where the display of nodes, elements, contours, and other items are controlled.

### 10.10 Model Check

A *Model Check* should be performed on all *HIVEL* models before attempting an analysis. The model check will perform a basic check to insure that all of the needed information to run the analysis is present. The *Model Check* command in the *HIVEL* menu causes the *Model Check* dialog (see Figure 10.8) to appear.

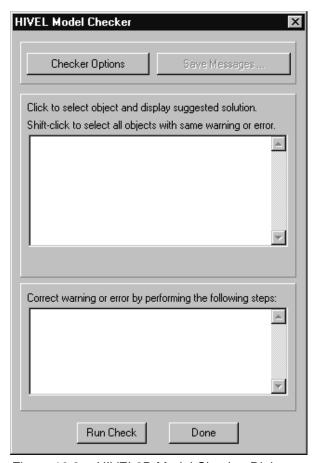


Figure 10.8 HIVEL2D Model Checker Dialog

Selecting the *Checker Options* button will cause the *HIVEL Model Checking Options* dialog to appear. This dialog lists the checks that may be performed during the m model checking procedure. By default all supported checks are enabled. The checks include:

- Check boundary conditions. When this option is turned on, SMS will make sure inflow and outflow have both been defined. It will also check to see that the outflow specification is higher than the highest node elevation.
- Check model controls. Parameters that are checked when this option is turned
  on are the of turbulence coefficients, Petrov-Galerkin weight coefficients,
  and backwards differencing used. See section 10.7 for more information on
  these coefficients.
- Mesh Check Options. Generic model checking options include: renumbered
  mesh, material definition, duplicate nodes and small voids or small
  unexpected interior mesh boundaries. In this dialog you can also choose to
  show all or limit the number of similar messages to report.

Click the *Run Check* button to perform a model check with the currently selected options. After *SMS* has completed its model check, the error and warning messages

will appear in the top text window. Some of these messages will include directions on how to fix it. To read these directions in the lower text window, click on the message. To save this information to a text log file, click the Save Messages button and enter a file name. To close the Model Checker, click the Done button. The model checker may remain open while changes are made to the mesh.

# 10.11 Run HIVEL

This option is not available on Win3.1 or Win3.11 for Workgroups. It is supported on all other platforms. This starts the HIVEL model to run a solution on your current data. If the current data has been edited since it was last changed, you will be prompted to save it first.

CHAPTER 11

# FESWMS Interface

FESWMS is a hydrodynamic modeling code that supports both super and subcritical flow analysis, including area wetting and drying. It has been developed under funding by the U.S. Federal Highways Administration (FHWA) by Dr. Dave Froelich P.E.. FESWMS is specifically suited for modeling regions involving flow control structures, such as are encountered at the intersection of roadways and waterways. Specifically, the FESWMS model allows the user to include weirs, culverts, drop inlets, and bridge piers into a standard 2D finite element model. SMS provides graphical tools for defining these structures and controlling analysis using the FESWMS model. Both pre- and post-processing capabilities are included in the interface.

The *FESWMS* version 2.x package consists of three programs: DIN2DH, FLO2DH, and ANO2DH. Earlier versions of *FESWMS* referred to these programs as DINMOD, FLOMOD, and ANOMOD. DIN2DH and ANO2DH are non-interactive programs for mesh generation and plot generation, respectively. When using *SMS*, only the FLO2DH program is used. The FLO2DH program is the analysis engine of *FESWMS*. *SMS* supports both version 1.x and 2.x of the *FESWMS* analysis package.

This chapter describes the commands used to create and edit the *FESWMS* specific boundary conditions, flow control structures, run parameters, etc., included in the *FESWMS* menu. The commands for generating and editing the mesh are described in Chapter 4. Once the analysis is complete, post-processing of the analysis results is performed by importing the solution file into the *Data Browser*. (See Lesson 6 of the *SMS Tutorial* and the *FESWMS User Manual* for more about running FLO2DH).

# 11.1 Open Simulation

The *Open Simulation* command reads in a FLO2DH file, which contains a mesh that has been previously created and saved. File I/O for a *FESWMS* problem is controlled through a project file which typically has the file extension *fil*. This file contains the names of ten other files that may or may not be required for an analysis. The order these filenames appear in the project depends on the version of *FESWMS* being used. The order for each version is:

FLOMOD (FESWMS version 1.x)	FLO2DH (FESWMS version 2.x)
Control data file - ".dat"	Control data file - ".dat"
Printed output file - ".pnt"	Grid network file - ".net"
Grid network file - ".net"	Initial flow file - ".ini"
Initial flow file - ".ini"	Boundary condition file - ".bnd"
Boundary condition file - ".bnd"	Wind data file - ".wnd"
Wind data file - ".wnd"	Printed output file - ".pnt"
Solution output file - ".out"	Solution output file - ".out"
Restart/recovery file - ".rsr"	Restart/recovery file - ".rsr"
upper matrix decomposition file - ".upp"	upper matrix decomposition file - ".upp"
lower matrix decomposition file - ".low"	lower matrix decomposition file - ".low"

Whether a particular file is required depends on the options specified by the user and stored in the control data file. *SMS* has been designed to easily control these options. As a new FLO2DH file is read, any previous mesh data is deleted from memory. If this data has not been saved since it was last edited, a warning message will appear. The name of the current FLO2DH file will be displayed at the top of the *Main Graphics Window*.

The files that are (or can be) used by FLO2DH or FLOMOD include:

- Control data file ".dat" contains run control parameters and data.
- Grid network file ".net" lists nodes and element connectivity's.
- Initial flow file ".ini" specifies initial flow conditions. Same format as a solution file created by *FESWMS*.
- Boundary condition file ".bnd" specifies the flows, water surface elevations etc. at mesh boundaries
- Wind data file ".wnd" specifies wind conditions.
- Printed output file ".pnt" specifies name of the file created by *FLO2DH*.
- Solution output file ".out" specifies name of solution file to be created. Contains output data at each iteration or time step as specified by the user.

Solution files contain velocities and water surface elevations at each node. For dynamic solutions derivatives of these quantities are also specified, and a solution at each time step is recorded.

- Restart/recovery file ".rsr" solution written at a specified interval for hotstarting future runs. (We recommend that this file not be used)
- upper matrix decomposition file ".upp" used to compute the solution.
- lower matrix decomposition file ".low" used to compute the solution.

The formats of these files are discussed in the *FESWMS Users Manual*. All input data may be contained in the control file. The optional files exist to allow the user to break up the data for multiple variations of a problem. The control file includes a card which specifies which of the other files are being utilized by *FLO2DH*. To specify the use of optional files, see the *FESWMS CONTROL* section later in this chapter.

# 11.2 Save Simulation

The Save Simulation command causes the Save FESWMS dialog to appear. Once the user has selected the desired filenames, SMS saves the FLO2DH/FLOMOD project file (.fil) and all of the other input files that the user has specified in the FESWMS CONTROL dialog. The current directory and project file is shown in the top of the Save FESWMS dialog (see Figure 11.1). The directory may be changed by pressing the FLO2DH .fil button and choosing a directory. The user should enter a desired filename in the .fil edit box and push the Update button to automatically give all files the same prefix. If desired, however, each filename may be entered separately. When all the filenames are entered, push the OK button. If the .fil file already exists, a prompt will appear. However, no check is made on the existence of other files.

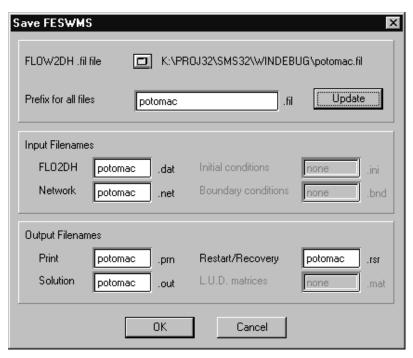


Figure 11.1 Save FESWMS Dialog

# 11.3 Nodal Boundary Conditions

The Assign BC command is used to specify boundary conditions at selected nodes or nodestrings. Nodal boundary conditions are assigned if the Select Node tool has been selected. Flow rates and water surface elevations may only be specified at nodes on the mesh boundary. However, a water source or sink may be specified at any node in the mesh. Before choosing this menu item, at least one node must be selected. In the FESWMS Nodal Boundary Conditions dialog (see Figure 11.2), the user may select any combination of boundary conditions.

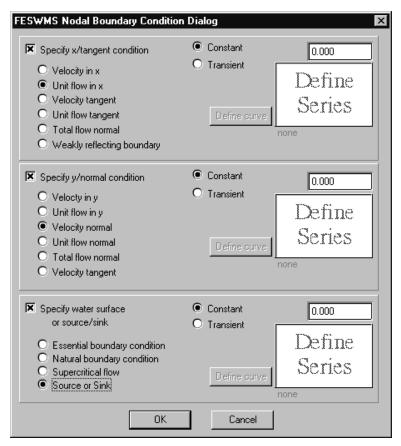


Figure 11.2 FESWMS Nodal Boundary Condition Dialog.

## 11.3.1 Specify x/Tangent Condition

The first type of nodal boundary condition is the velocity or flow in the x direction or tangent to the boundary. The options are as follows:

- Velocity/unit flow can be specified with respect to the positive x-axis.
- Velocity/unit flow can be specified *tangent* to the mesh boundary at the node.
- Total flow *normal* to the *open* boundary at that node can be specified.
- Weakly reflecting boundary invariant is a constant used to indicate the relationship of flow between outflow and tidal cycles.

## 11.3.2 Specify y/Normal Condition

The second type of nodal boundary condition is the velocity or flow in the y direction or normal to the boundary. The options are as follows:

- Velocity/unit flow can be specified with respect to the positive y-axis.
- Velocity/unit flow can be specified *tangent* to the mesh boundary at the node.
- Total flow normal to the closed boundary at that node can be specified.
- Velocity *tangent* to the *Open* boundary at that node can be specified.

## 11.3.3 Specify Water Surface or Source/Sink

If the selected node is on the mesh boundary, the water surface elevation or supercritical flow can be specified at the node. Otherwise, the node is on the interior of the mesh and only a source or sink is allowed. Water surface elevation can be specified as either an essential boundary condition or a natural boundary condition. If the water surface elevation is essential, *FESWMS* will not allow the value to fluctuate at all. If it is natural, small fluctuations are allowed. If supercritical flow is chosen, no water surface elevation value is entered. If a sink exists at the node, a negative value should be entered. Otherwise, a positive value should be entered.

# 11.4 Boundary Section

The Assign BC command is used to specify boundary conditions at selected nodes or nodestrings. Nodestring boundary conditions are assigned if the Select Nodestring tool has been selected. Boundary Sections are used for specifying the conditions existing at an open boundary. Inflow, outflow, and water surface elevation may be specified on a selected nodestring which lies on the mesh boundary. The conditions along the boundary are specified using the Boundary Section dialog (see Figure 11.3).

#### 11.4.1 Flow

Flow defined with a nodestring at an open boundary is always considered to be perpendicular to the mesh boundary when using *FESWMS*. However, it can be defined as an inflow or an outflow. An inflow is defined by positive values, while an outflow is defined with negative values. *FESWMS* version 2.x allows the user to

specify flow either as direct flow or as a flow across a weakly reflecting boundary for tidal situations (see the FESWMS version 2.x reference manual). If a user specifies a weakly reflecting flow boundary, and saves the data as a FESWMS version 1.x file, the boundary condition is converted to a regular flow boundary condition.

#### 11.4.2 Water Surface Elevation

As with nodal boundary conditions, water surface elevation defined at a nodestring is defined as either essential or natural. The water surface elevation option also allows for the definition of supercritical flow existing at the boundary, for which no values will be defined.

If either an essential (no fluctuation allowed) or natural (small fluctuations allowed) water surface elevation is chosen, the value can be constant or it can vary across the boundary. If the water surface elevation varies, initial and end values are specified. From these values the water surface elevation will be interpolated to each node of the nodestring.

Water surface elevation may also vary along a string or be supercritical.

## 11.4.3 Rating Curve/Friction Slope

The *Rating curves* and *Friction slope* items in the *FESWMS Nodestring Boundary Conditions* dialog should not used for *FESWMS* version 1.x. A rating curve describes the relationship between water surface elevation and flow rate, and may be defined at a nodestring which has an essential or natural water surface elevation defined. This means that the water surface elevation at each node of the nodestring will depend on the flow at that node. Up to eight values can be entered for a specific rate curve. If the friction slope option is chosen, the water surface elevation applied to each node of the nodestring will be calculated using the slope-area method.

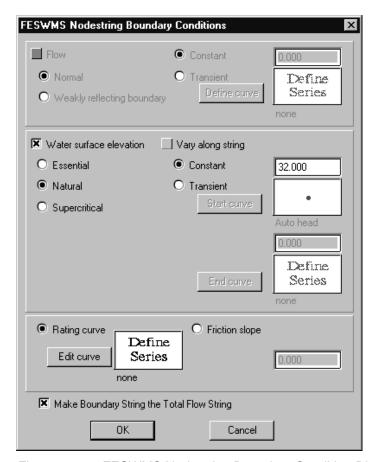


Figure 11.3 FESWMS Nodestring Boundary Condition Dialog.

## 11.5 Initial Conditions

SMS currently does not support an interface for specifying initial conditions. If these are desired for a mesh, an initial condition file must be created with a text editor. The format of this file can be found in the FESWMS users manual. If such a file is created, be sure that the initial condition flag is set to 1 in the model control file and that the initial condition filename is in the correct position in the .fil file. (See Open , Section 11.1).

## 11.6 Wind Conditions

Global wind conditions may be applied to the model though the *FESWMS Wind Conditions* dialog (see Figure 11.4). If a steady-state solution is being performed, only constant wind values can be entered. Time variant or constant values can be entered if a time-dependent solution is to be performed. (See FESWMS Control, section 11.15 for more about changing the solution type).

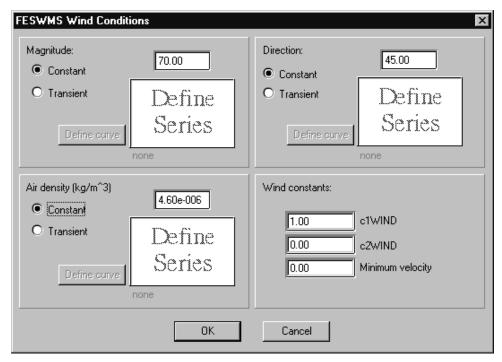


Figure 11.4 FESWMS Wind Dialog.

The units of wind magnitude and air density depend on the units being used (SI or English). The wind direction is entered in degrees counter-clockwise from the positive x-axis. The three wind constants are used in computing the stress coefficient at the water's surface (see the *FESWMS* users manual).

As with nodal initial conditions, *SMS* is currently not supporting an interface for nodal wind conditions. However, a wind file may be created using the format described in the *FESWMS* users manual. As with the initial condition file, the wind file flag should be set in the model control file, and the correct filename should be specified in the *.fil* file.

## 11.7 Weirs

Weirs can be defined on the boundary or interior to the mesh. One or two nodes must be selected before defining a weir. If the water which flows over the weir does not return to the mesh, only the upstream node is needed. However, if the water is to return to the mesh, two nodes should be assigned to the weir.

The upstream node should have a higher water surface elevation than the downstream node, as water flows over a weir only in the direction from the upstream node to the downstream node. Also, the weir's crest elevation should be lower than the upstream node's water surface elevation, or water will not be able to flow over the weir.

After a weir is defined, it is displayed according to the defined attributes (see page 11-20 to learn how to change the display). If two nodes are used to define a weir, a line is drawn in the weir attribute color from the upstream node to the downstream node.

## 11.8 Culverts

Culverts can be defined on the boundary or interior to the mesh. One or two nodes must be selected before trying to assign a culvert. If the water which flows through the culvert does not return to the mesh, only the upstream node is needed. However, if the water is to return to the mesh, two nodes should be assigned to the culvert.

The geometry of the culvert is defined by the cross sectional area and barrel length. Entrance loss, hydraulic radius, and Manning's roughness coefficient establish how well water flows into and through the culvert.

If the flap-gate option is chosen, the culvert will be treated as though only the upstream node was selected. In this case, water will leave the mesh but will not return.

# 11.9 Drop Inlets

Drop inlets can be defined on the boundary or interior to the mesh. One or two nodes must be selected before trying to assign a drop inlet. If the water which flows over and through the drop inlet does not return to the mesh, only the upstream node is needed. In this case, the hydraulic head elevation at the drop inlet spillway should be specified. If the water is to return to the mesh, two nodes must be assigned to the drop inlet.

The weir parameters are used for computing flow at the opening of the drop inlet if it is not fully submerged. If the entrance is submerged, entrance flow will be computed using the orifice parameters. Flow inside the drop inlet is computed using the conduit parameters. All parameters are necessary except for the hydraulic energy head.

## 11.10 Piers

To define piers, a mesh must already exist. In the *Piers* dialog, the pier coordinates, drag coefficient, and shape factors are defined. To add a pier, either click the *Add Pier* button or, if any field of the last pier is highlighted, hit the *DOWN ARROW* key. To delete a pier, select any cell on the pier's row and click the *Delete Pier* button.

# 11.11 Node Ceilings

Nodes can be assigned to have a ceiling value, or a value above which water flow is prohibited. This is useful for modeling river flow under a bridge. If the water surface is above the ceiling value at a specific node, pressure flow will result. Otherwise, flow at the node is modeled as free-surface flow.

To assign a ceiling value the user selects the nodes for which a ceiling is desired and selects the Ceiling command from the FESWMS menu. SMS will then prompt for the ceiling value.

SMS creates a data set for the ceiling elevations when a FESWMS file is input or the node ceiling values are edited. This data set can be selected through the data browser (see Section 3.1) and viewed as contours. This facilitates visualization of the ceiling values.

# 11.12 Flux String

A flux string is a nodestring at which continuity checks will be made during a solution run. Flux strings are generally placed at cross sections of the model. To define a flux string, at least one nodestring must be selected. Then the user selects the FLUX String command from the FESWMS menu. The selected nodestring(s) will be assigned as a flux string(s). The color of flux strings can be set in the Continuity strings option of Nodestrings options of the Display Options dialog.

# 11.13 Material Properties

FESWMS allows the user to set model specific material properties using the FESWMS Material dialog (see Figure 11.5). The user can specify the color, material name, and other general material parameters using the General Materials dialog.

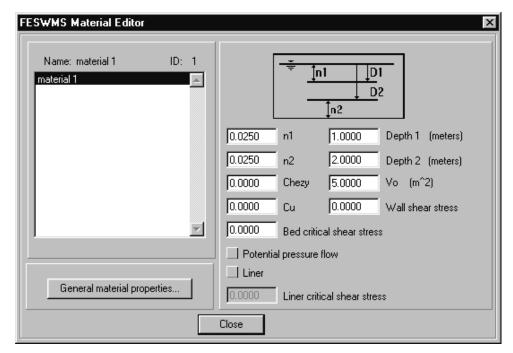


Figure 11.5 FESWMS Material Editor Dialog.

#### D1, n1, D2, n2

FESWMS supports manning roughness coefficients that vary with depth. The first roughness coefficient (n1) will be applied to all water depths less than the first depth (D1). The second roughness coefficient (n2) will be applied to all water depths greater than the second depth (D2). If a value of zero is entered, n1 will be applied to the entire system.

#### **Other Coefficients**

Other coefficients are applied to the entire model regardless of water depth. The Chezy discharge coefficient is used to calculate the bed friction coefficient. Vo and  $C\mu$  are used to calculate eddy viscosity.

#### **General Material Properties**

In the general *Materials* dialog, materials are created and deleted. Display colors of each material is set, as is the material name. After materials are created by using this option, they can be edited in the *FESWMS Material Editor*.

## 11.14 Model Check

The FESWMS Model Checker will check the mesh, making sure that all elements have material properties assigned and are properly defined, as well as checking items specific to FESWMS. By selecting the Checker Options button, items to be checked

may be selected, both for *FESWMS* input data and for geometry data. If the *Check Boundary Conditions* option is chosen, the existence of boundary flow and water surface elevation will be checked. The *Check Water Surface Elevation* option will check if all defined water surface elevations are higher than the lowest nodal elevation.

After desired options are selected, press the *Run Check* button to check the current mesh. After a mesh has been checked, error messages may be saved by clicking on the *Save Messages* button. This dialog may be kept open while the mesh is edited, or it can be closed. (Section 8.10 discusses and illustrates the operation of the *Model Check*).

## 11.15 FESWMS Control

The user can specify what options will be used for the numerical analysis through the *FESWMS Control* dialog (see Figure 11.6).

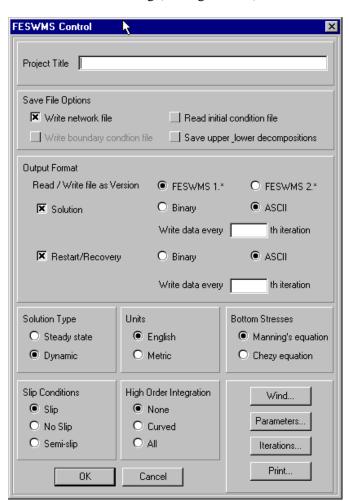


Figure 11.6 FESWMS Control Dialog.

## 11.15.1 Project Title

The *Project Title* allows the user to name the problem under consideration. It is saved with the control data file, and is written at the top of the *Main Graphics Window*.

## 11.15.2 Save File Options

These *Save* options tell *SMS* which files should contain specific portions of the *FESWMS* data. These flags are saved in the control data file and tell *FESWMS* where to find the data to run a solution. If none of these options are selected, all mesh data will be written in the *FESWMS* control data file (see the *Open FLO2DH* section). The *Write Network File* option tells *SMS* to save the nodes and element connectivities to a separate grid file. The *Write Boundary Condition File* tells *SMS* to save boundary conditions to a separate boundary condition file. Although *SMS* does not currently support an interface for creating initial condition files, if one will be supplied, the *Write Initial Condition File* option should be selected. The *Save Upper and Lower Decompositions* option allows for files containing upper and lower decompositions of the coefficient matrix to be saved at the end of a solution run instead of being deleted.

If any of these options are specified, the filenames should be entered in the *Save FESWMS* dialog, as described in the *Save FLO2DH* section of this chapter.

## 11.15.3 Output Format

The *File Version* corresponds to the version of *FESWMS* which will be used for running a solution. Mesh data should be saved accordingly. If the solution option is chosen, *FESWMS* will output the solution data to a file in either ASCII or binary format. If a time dependent analysis is being performed, solution data will be written at specified time steps.

A restart/recovery file can also be written while a solution is being performed. In the event that the program is interrupted, this allows the solution to continue where it left off. The solution is continued by using the restart/recovery file.

## 11.15.4 Solution Type

The solution type is specified as either steady state or time-dependent. If a steady-state solution is to be performed, all boundary conditions such as wind, flow, and water surface elevation should be defined as constant. If a time-dependent solution is to be performed, all boundary conditions may be either constant or time-dependent.

#### 11.15.5 Units

The units for the model are specified as English or metric. Metric units are the default. If the units are changed, gravity constants and air densities are converted. However, initial conditions are not converted. It is a good idea to check all specified values if the units are changed after the values have been defined.

#### 11.15.6 Bottom Stresses

Either Manning's Equation or the Chezy Equation is specified for analyzing shear stresses at the water bed. The differences between these two algorithms are described in any basic fluid mechanics book.

## 11.15.7 Slip Conditions

A slip condition will be specified for all closed boundaries during the run. If Slip is chosen, no shear stress will be applied to closed boundaries. No Slip indicates that the shear stress at closed boundaries is so great that tangential velocity is zero. Semi-slip allows for slip at a closed boundary unless the flow is against a vertical wall. If the semi-slip option is chosen, a value should be entered in the Vertical wall shear coefficient in the FESWMS Parameters dialog.

## 11.15.8 Higher Order Integration

Low order integration is used as the default for all calculations unless otherwise specified. The Curved option tells FESWMS to use high-order integration only when dealing with curve-sided elements. The All option tells FESWMS to always use highorder integration. Using high-order integration gives more precise solutions, but also takes longer to perform the analysis.

#### 11.15.9 Control Buttons

#### Wind Dialog

The Wind button opens up the FESWMS Wind dialog. (See Section 11.15.9 for information about this dialog).

#### **FESWMS Parameters**

The Parameters button causes the FESWMS Parameters dialog (see Figure 11.7) to appear. This dialog allows a user to edit optional run control parameters.

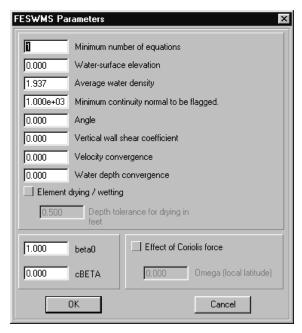


Figure 11.7 FESWMS Parameters Dialog.

Optional parameters for *FESWMS* that are specified in the *FESWMS Parameters* dialog include:

#### Minimum number of equations

The minimum number of equations value is the number of completed equations retained in the active solution matrix during a frontal solution run. This value is ignored for *FESWMS* versions 2.x. The default value is 1.

#### Water-surface elevation

The default water-surface elevation (wse) entered will be assigned to all nodes which do not have an assigned boundary wse. The units of wse depend on the solution type and should be automatically converted if the unit base is changed. When using English units, enter a value in feet; for SI units, enter a value in meters.

#### Average water density

Average density of the water flowing through the mesh should be entered according to the solution units (English or SI). The current units are displayed next to the input field.

#### Minimum continuity normal to be flagged

The *Minimum continuity normal to be flagged* data entry allows for the marking of continuity norms greater than the value indicated. Any such continuity norms will be marked with an asterisk in the printed output file. The default value is 1.0e35, meaning continuity norms will not be flagged in the file.

#### Angle

This data entry field defines the angle between the positive x-axis and true north. It should be given in degrees clockwise from true north. For example, if true north is along the positive y-axis, a value of 90 should be entered.

#### Vertical wall shear coefficient

If a semi-slip condition is specified and the shear at a vertical wall is the same as the bed shear, this value should be specified as zero. Otherwise, the shear coefficient value should be entered.

#### Element drying/wetting

The Element drying / wetting toggle is used to control how FESWMS handles changing water surface elevation. If this toggle is not selected, any element that is not fully submerged during a solution will be excluded from the analysis. If it is selected, the specified tolerance will be used to determine if a partially submerged element should or should not be included in the analysis. If the numerical model determines that an element has dried out, the flow that was passing through that element is passed over to elements that are still wet. Therefore, if too many elements get classified as dry in a single iteration, the water surface elevations in the remaining elements may go up enough to incorrectly make the dry elements wet again. This shifting back and forth leads to numerical instability in the solution. By specifying the Depth Tolerance, the elements will dry out in a more distributed fashion, minimizing the risk of instability.

#### beta0 and cBETA

These coefficients are used for computing the momentum flux correction coefficients. Default values are Beta0 = 1.0, cBETA = 0.0. The momentum flux correction coefficients are used to determine the change in water velocity with change in depth. The default values mean that such velocity changes are negligible, which is generally the case except in very shallow water.

#### Effect of Coriolis force

If an angle other than 0 is entered, the effect of the Coriolis force will be considered. The angle is the average latitude in degrees of the model. A positive angle should be entered if the water body is in the Northern Hemisphere, and negative if it is in the Southern Hemisphere.

#### FESWMS Iterations

The number of iterations used during the analysis is controlled via the FESWMS Iterations dialog (see Figure 11.8) which is invoked from the FESWMS Control dialog.

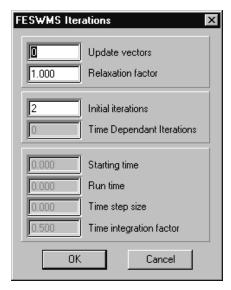


Figure 11.8 FESWMS Steady-State Solution Iterations Dialog.

The *Number of initial iterations* is the number of Newton iterations initially performed during a solution run. We highly recommend that the user specifies all iterations in the *Number of initial iterations* box. After these are performed, the run will continue until the solution converges or until the *Number of Additional Full Newton iterations* have been performed.

The *Relaxation Factor* is used by *FESWMS* to scale the correction at the end of each Newton iteration. The default value is 1. If a model demonstrates slow convergence, it may be sped up by increasing this factor (up to 2). This should be used with caution however, as increasing the factor may lead to instability. Likewise, if a model diverges, the relaxation factor may be decreased to increase the numerical stability. A lower value will give a slower but more stable solution run.

If a time dependent solution is being run, additional fields are provided to specify time dependent parameters. These are the starting time, run time, time step size and time dependent iterations. A time integration factor (between 0 and 1) should also be specified. The default integration factor is 0.5.

#### **FESWMS Print**

The *Print* button at the lower right corner of the *FESWMS Control* dialog brings up the *FESWMS Print Options* dialog (see Figure 11.9).

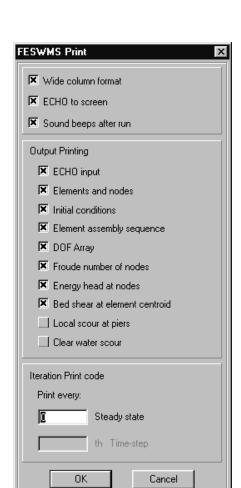


Figure 11.9 The FESWMS Print Dialog

#### Wide Column Format and ECHO to Screen

Wide column format means that the output will be printed in 132-column format. If this option is not selected, output will be printed in 80-column format. Echo to screen indicates that as a solution is run, messages will be written to the screen describing the status of the solution. Otherwise, such messages will not be seen.

#### **Output Printing Options**

When a solution is run, an output file is created. This output file contains control data, error messages, and solution results. Options in this section control other data which may or may not be printed. For example, if *ECHO Input* is selected, everything read from data records will be echo printed. If *Froude Number at Nodes* is selected, the Froude number for each node will be calculated and printed at each specified iteration/time step (see *Iteration Print Code* in the next paragraph).

#### Iteration print code

These fields specify how often the *Output Printing Options* will be written to the printed output file. For a Steady-state solution, only an iteration interval can be

specified. For a Time Dependent solution, output printing is based on iteration intervals and specified time step values.

# 11.16 FESWMS Display Options

As with the general display options, the display of items specific to *FESWMS* can be changed using the *Display Options* command in the *FESWMS* menu. The symbol and color of the items can be changed in the *Display Options* command. *FESWMS* specific display options include:

- The *Nodestring BC Display Options* item controls the symbol and color for display of boundary conditions applied to nodestrings.
- The General Display Options item brings up the Display Options command.

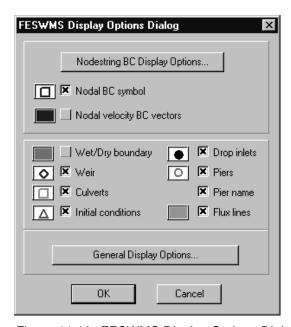


Figure 11.10 FESWMS Display Options Dialog

CHAPTER 12

# WSPRO Interface

WSPRO is a water-surface profile computation model that can be used to analyze one-dimensional, gradually-varied, steady flow in open channels. WSPRO also can be used to analyze flow through bridges and culverts, embankment overflow, and multiple-opening stream crossings. WSPRO is supported and maintained by the U.S. Federal Highways Administration (FHWA).

WSPRO categorizes its input data into five general groups:

- *Title information*. Used for output identification.
- *Job parameters*. Used to define parameters that pertain to the entirety of the profile computations.
- *Profile control data*. Information regarding discharge, starting elevation and computation direction.
- *Cross-section definition*. Information describing physical system (geometry, roughness, etc.).
- *Data display commands*. Used to control output of tables of cross-sectional properties, velocity, conveyance, etc..

*SMS* provides graphical tools for defining and editing the data in each of these groups, graphically editing cross-sectional information, and visualization of profiles and cross-sectional properties computed by *WSPRO*.

This chapter describes the commands used to create and edit the WSPRO specific parameters included in the WSPRO menu. The commands for selecting sections and

operating on the river model are described in Chapter 7. SMS also interfaces to WSPRO to invoke analysis of a river model. After the analysis is complete, SMS can import the solution file via the WSPRO Display Options Dialog to allow the user to view cross section and profile plots of all the data generated by WSPRO. (See Lesson 13 of the SMS Tutorial and the WSPRO User Manual for more about running WSPRO).

## 12.1 New Simulation

The *New Simulation* command in the *WSPRO* menu deletes the current river model including all of the *WSPRO* specific data. This data includes the run control data, the job parameters, and the solution data. Run control data contains the computation directions, profile discharges etc. Job parameters include output tables and tolerances. Solution data includes all variables computed for profile or cross section visualization that is not geometric. The *New Simulation* command also deletes the general river model data(geometric cross sections). To delete all the data currently in *SMS*, the user should select *New* from the *File* menu which causes all existing data (geometry and model specific data) to be deleted from memory.

# 12.2 Open Simulation

The *Open Simulation* command in the *WSPRO* menu reads in a data file that has been previously created and saved. These files typically have the file extension ".dat". The name of the current simulation is displayed at the top of the *River Window*. The data file contains both the geometric and model control data for a *WSPRO* analysis. Geometric data consists of both the definition of the shape of the section and the section reference distance locating sections in relation to each other. (See the *WSPRO User Manual* for more information of data file formats). Opening a new simulation file causes all existing river data (geometry and model specific data) to be deleted from memory, however, data in other formats (such as two dimensional mesh data is not affected).

## 12.3 Save Simulation

The *Save Simulation* command in the *WSPRO* menu saves a data file so that it can be opened at a later time or used in an analysis.

## 12.4 Edit Section

The *Edit Section* command in the *WSPRO* menu invokes the *Section Editor* (see Figure 12.1). This editor can also be invoked by double clicking on a section in the *River Window*.

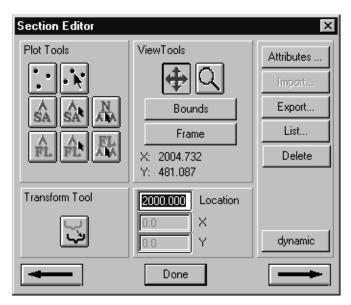


Figure 12.1 WSPRO River Section Editor.

The Section Editor is divided into five areas. These include the plot tools in the upper left portion of the dialog, the view tools in the upper center, the section translation tool in the lower left, section edit fields in the lower center, and various graphical buttons on the right side. Each of these areas are described in some detail in the sections to follow. In the bottom corners of the Section Editor are arrows that allow the user to step through the selected sections and edit each in turn. If only one section is selected, the arrows cause the current section to be unselected and the next section in the river model to be selected.

#### 12.4.1 Plot Tools

The plot tools area in the upper left portion of the *Section Editor* presents a set of tools that apply to the river section that is currently being edited. For example, if the user is editing a cross section, up to three rows of tools are available (see Figure 12.1). The first row consists of tools to edit the geometry points (GR records). The second row consists of tools to create and edit the materials data including break points (SA records) and roughness data (N or ND records). The third row includes tools to specify flow lengths (FL records) for different portions of the section. The visible tools change based on the section type, and the section position. For example, the flow length tools are not visible if the section being edited is the outlet of the river model, due to the fact that no flow lengths are applicable to the outlet. They also

vanish if the section has a bridge section defined because this is an unsupported combination of records in WSPRO.

The tools for bridges, guide banks, culverts and roads are all specific to the section type. The available tools also change based on the attributes of the section. An instance of this is a component mode bridge as opposed to a coordinate mode bridge. Coordinate mode includes tools for geometry point manipulation, while component mode does not.

#### 12.4.2 View Tools

The view tools area in the upper center portion of the *Section Editor* presents tools which allow the user to pan, zoom, frame, and specify specific window boundaries for the plots of the cross sections and profiles. These tools function exactly as their counterparts in the display menu for the graphics window (see sections 2.3.2 & 2.9.4).

*SMS* tracks the location of the cursor in the plot window, both in cross sections and profile plots. This information is reported below the view tools.

# 12.4.3 Section Translation Tools

The section translation tool is located in the lower left portion of the *Section Editor*. When this tool is selected, the user can translate the entire section either graphically by clicking on the cross section and dragging the geometric data. This is useful for general placement of one section with respect to another. For example, a bridge can be dragged to an approximate location with respect to the full valley cross section.

If a specific translation is desired, such as would be the case if the datum for one section was different than the other sections in the model, the user can specify the translation in the *X* direction and/or the *Y* direction and click on the bottom graphics button on the right side of the editor to invoke a translation. This bottom button reads "translate" while the section translation tool is selected.

#### 12.4.4 Section Edit Fields

The lower middle region of the *Section Editor* includes three edit fields that are used to enter specific numbers for editing the section. The top edit field allows the user to specify a new section reference distance (SRD) for the current section. The lower two edit fields change function based on the current tool. For example, if the current tool is the edit geometry point tool , the edit fields represent the X and Y location of the currently selected geometry pt (GR record). This allows the user to edit the point graphically in the plot window, or explicitly using the edit fields. These fields are also used for specific section translations as described in Section 12.4.3, as well as

specific values for SA or FL breakpoints for the section. The label to the right of the edit field updates to reflect the current purpose of the edit field.

### 12.4.5 Section Editor Buttons

The right side of the *Section Editor* includes six buttons to operate on the section. The *Attribute* button allows the user to access additional dialogs to edit the current section. Each section type has its own set of attributes. For each section type, the attribute dialog is described below.

#### **Cross Section Attributes**

Cross sections require only a name and section reference distance (SRD). The SRD is specified in the *Section Editor*, the name can be specified at the top of the *Cross Section* dialog. The user can access other optional attributes including skew, loss coefficients, valley slope and friction slope averaging method. Default values are filled into these attributes when they are turned on. Any unspecified attribute is entered as a blank in the *WSPRO* data file, causing *WSPRO* to assume a default value.

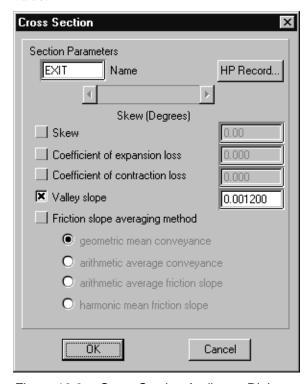


Figure 12.2 Cross Section Attributes Dialog.

WSPRO also includes an option to create tables of cross sectional properties (HP record). The user defines the desired tables using the HP Record button in the upper right corner of the attribute dialog. This invokes the WSPRO HP Tables dialog (see Figure 12.3). Tables can be generated for the entire cross section or for sub areas. The user may also define one velocity conveyance table for each section. If the user

desires additional velocity conveyance tables the extra records must be managed by hand.

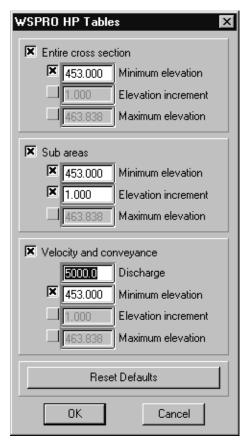


Figure 12.3 HP Record Dialog.

## **Bridge Section Attributes**

Bridge sections have many attributes. The *Bridge Section* dialog is divided into four parts.

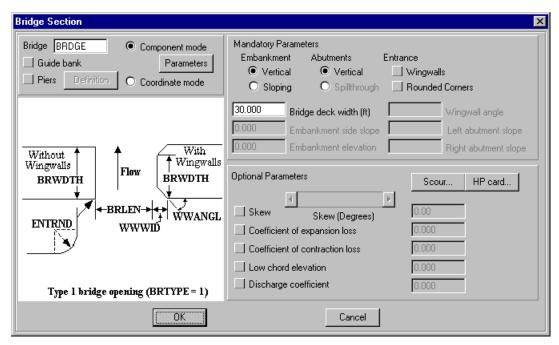


Figure 12.4 Bridge Section Attributes Dialog.

The upper left portion allows the user to change the section name, add piers and/or guidebanks, and specify the geometry mode for the bridge. WSPRO allows bridges to be defined explicitly (coordinate mode) using points (GR record), or explicitly using parameters (component mode). If component mode is used, the parameters defining the bridge are defined using the Bridge Component Mode Parameters dialog (see Figure 12.5), which is invoked using the Parameters button. If the Pier toggle is selected, SMS will create a pier record for the bridge (PD record). The Definition button allows the user to define the widths of the piers and the number of piers at different elevation.

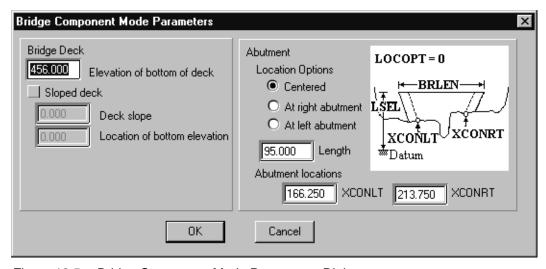


Figure 12.5 Bridge Component Mode Parameters Dialog.

The upper right portion of the dialog includes the mandatory parameters for the bridge. These include the embankment type, the abutment type, the deck width and several slopes and angles that may or may not be required depending on the bridge type and mode. The elements which don't apply to the currently selected type are dimmed. The lower left portion displays the parameters for the currently selected bridge type.

The lower right portion includes optional parameters including skew, discharge and loss coefficients., cross sectional, and scour parameters. The scour is defined using the *Bridge Scour Options* dialog (see Figure 12.6). Three types of scour records are defined. These include abutment scour (DA records), live-bed/clear-water scour (DC records), and pier scour (DP records).

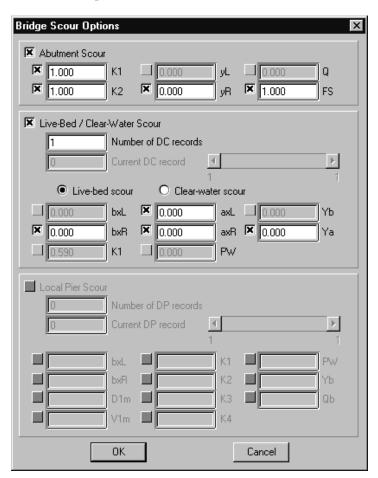


Figure 12.6 Bridge Scour Options Dialog.

#### **Road Section Attributes**

Road grade sections require a name, SRD, road type and embankment top width. Optional parameters include a skew and weir flow coefficient. These are all specified in the *Road Grade Section* dialog.

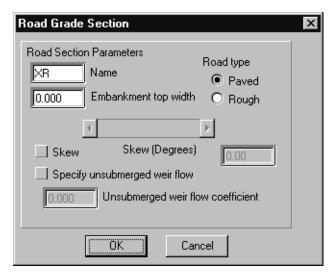


Figure 12.7 Road Section Attributes Dialog.

#### **Culvert Section Attributes**

Culvert sections can be used to model simple culverts, or culverts in combinations with one or more bridge openings. Culvert section attributes are specified using the *Culvert Section* dialog. The left side of the dialog allows for definition of culvert parameters, while the right side allows the users to define the type and size of culvert.

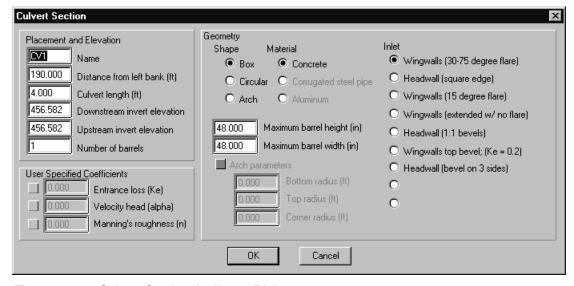


Figure 12.8 Culvert Section Attributes Dialog.

#### **Guide Bank Section Attributes**

Guide bank sections are bound to a specific bridge section. They are created by selecting the *Guide Bank* toggle in *the Bridge Section* dialog (see Figure 12.4). Once a guide bank exists, its parameters can be modified using the *Guide Bank Section* dialog accessed through the *Section Editor*. This dialog allows the user to define the

name of the section, loss coefficients, skew, cross sectional tables, and the type of guide bank.

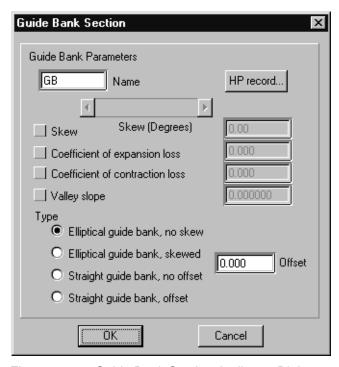


Figure 12.9 Guide Bank Section Attributes Dialog.

## 12.5 View Data File

The *View Data File* command in the *WSPRO* menu asks the user to select a text editor and then a file. The default editor is Notepad on PC systems and VI on UNIX systems. *SMS* then launches the specified text editor with the specified file. This allows the user to look at the input files used to run *WSPRO*, and the output files generated by *WSPRO*.

# 12.6 Roughness Parameters

WSPRO uses roughness parameters or Manning's n values to simulate the different types of bed conditions in the river. In the Section Editor (section 12.4), the user assigns materials to the different regions in a cross section. The WSPRO Material Editor allows the user to associate Manning's n values to the materials used by each section. Materials can use a single roughness value, or have a roughness that varies based on the depth of the flow. This simulates situations such as tall grass that resists flow (high roughness) until the flow reaches a depth at which the grass is pushed over and lays down.

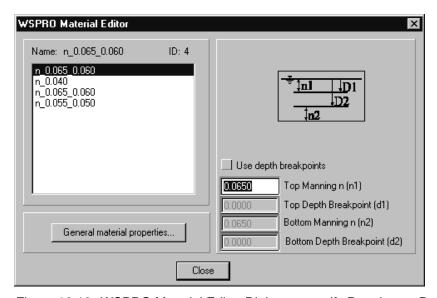


Figure 12.10 WSPRO Material Editor Dialog to specify Roughness Parameters.

## 12.7 WSPRO Run Control

WSPRO requires the user to specify the flow rates for which profiles are to be computed. The WSPRO Run Control item in the WSPRO menu allows the user to specify the flow rates, and boundary conditions as well as assign a title to the simulation. Multiple profiles can be computed at once, and branches in the river may be simulated by changing the flow rate at various sections. The user also controls the direction of computation. Upstream is usually used for subcritical situations while downstream is applied for critical and supercritical flow.

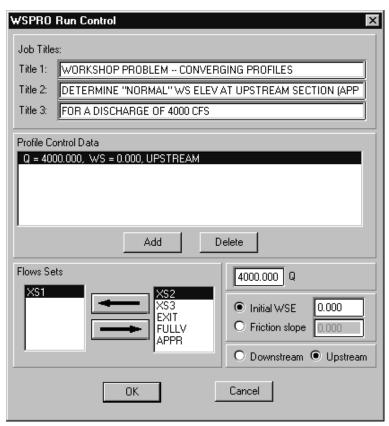


Figure 12.11 WSPRO Run Control Dialog

## 12.8 Job Parameters

The *Job Parameters* command allows the user to specify the optional parameters associated with a numerical analysis. These include computational step size and tolerances to be used during computation. The *WSPRO Job Parameters* dialog also allows the user to specify the units to be used in *WSPRO*, and to specify any output tables desired from the analysis.

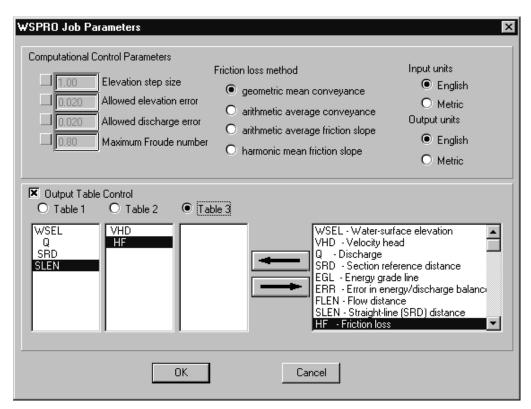


Figure 12.12 WSPRO Job Parameters Dialog

# 12.9 Display Options

SMS provides tools for both WSPRO model constructions and visualization of the results of a WSPRO analysis. The Display Options menu item allows you to specify what can be displayed, and how it will look. The top portion of the Display Options dialog allow the user to specify what colors will be used to display each type of section. The lower portion of the dialog includes tools to import the solution file created by WSPRO and specify which variables will be plotted in the Plot window.

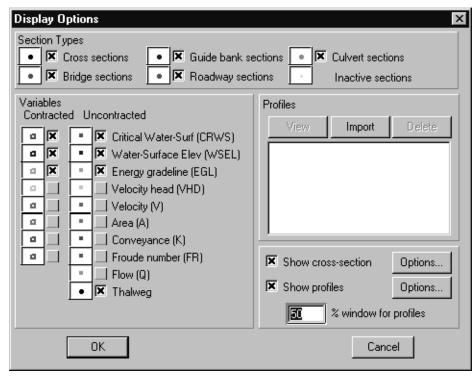


Figure 12.13 WSPRO Display Options Dialog

Each variable may be plotted for both the constricted and unconstricted flow cases. If no bridge or culvert sections exist, only the unconstricted case is available. The user also controls whether the plot window will include profiles, cross sections or both. If the user invokes the section editor, cross section display is turned on.

## 12.10 Model Check

A *Model Check* should be performed on all *WSPRO* models before attempting an analysis. The model check will perform a basic check to insure that all of the needed information to run the analysis is present. The *Model Check* command in the *WSPRO* menu causes the *Model Check* dialog to appear.

Selecting the Checker Options button will cause the WSPRO Model Checking Options dialog to appear. This dialog lists the checks that may be performed during the model checking procedure. By default all supported checks are enabled. The checks include:

- Check Profiles. This check assures that at least one profile has been specified and that the appropriate number of discharges and boundary conditions are specified for each profile.
- Check SRD Values. This option checks the SRD of each section in the model. SRD values should increase monotonically from the exit to the most upstream section. There are some exceptions for multiple opening situations. This option also checks the placement of bridge approach and exit sections to assure correct placement.
- *Check GR Points.* Cross sections geometry should proceed from left to right. Coordinate mode bridge openings should proceed across the bottom of the opening, then change direction one time and define the bottom of the bridge deck.

After running the model check, messages are generated to aid in the correction of the problems. To save this information to a text log file, click the Save Messages button and choose a file to save the information in. To close the Model Checker, click the Done button.

#### 12.11 Run WSPRO

Once the data for an analysis has been defined, WSPRO may be invoked by selecting the Run WSPRO menu item. This command checks the status of the model in SMS. If edits have been made, the user is prompted to save his files before running. It then launches WSPRO using the data files loaded into SMS. If multiple runs are desired, the user should go to the Display Options and delete previous solutions before reading in new one. numerical model requires several user specified parameters to control the analysis.

CHAPTER 13

# XY Series Editor

The XY Series Editor is a special dialog that is used to generate and edit curves defined by a list of x and y coordinates. The curve can be created and edited by directly editing the xy coordinates using a spreadsheet-like list of the coordinates. The curve can also be generated and edited graphically. An entire list of curves can be generated and edited with the Editor, and curves can be imported from and exported to text files for future use.

The XY Series Editor is used in several places in SMS. It was designed to be general in nature so that it could be used anywhere that a curve or function needs to be defined. In some cases, the x values of the curve must correspond to a pre-defined set of values. For example, the x values may correspond to a set of time steps whose interval is established in a separate dialog. In such cases, the x fields cannot be edited but the y values associated with the pre-defined x values can be edited. In other cases, there is no limit on the number of x values or on the x spacing and both the x and y values can be edited.

The XY Series Editor is shown in Figure 13.1. Each component of the dialog is described below.

#### 13.1 XY Series List

At the bottom of the dialog in the center, there is a list of xy series. One of the items in the list is active and highlighted at all times. The xy values of the active series are shown in the spreadsheet on the left side of the dialog and the curve is shown graphically in the upper right portion of the dialog. The name associated with the active series can be edited using the edit field to the right of the xy series list.

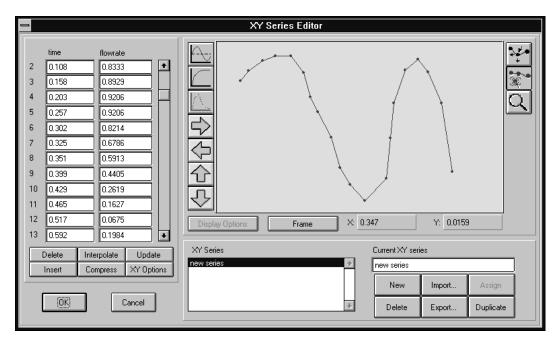


Figure 13.1 The XY Series Editor.

A new xy series can be created and added to the xy series list by selecting the *New* button. An existing series can be copied to create a new series by selecting the *Duplicate* button. This option is useful when two series need to be the same except for slight differences. An existing series can be deleted from the list by highlighting the series and selecting the *Delete* button to the right of the xy series list. A set of series can be read from a file by selecting the *Import* button. Likewise, the entire list of series can be saved to a file using the *Export* button. The file format used to save xy series is described in Section 14.14.

#### 13.2 The XY Edit Fields

The two vertical columns on the left side of the xy series dialog are for editing the values of the highlighted series. A title at the top of each column specifies what the points represent. The *TAB* key can be used to move the cursor through the edit fields.

If the number of points in the series is greater than the number of pairs of fields in the columns, the scroll bar to the right of the columns can be used to scroll through the entire range of the xy series.

The buttons below the xy edit fields are used to manipulate the values in the edit fields. A description of each button follows.

#### 13.2.1 Delete

The *Delete* button deletes the contents of the edit field that the cursor is located in. If the x field is static, only the y field is allowed to be deleted. Otherwise, both fields are deleted.

#### 13.2.2 Interpolate

The *Interpolate* button causes any blank fields in the xy series to be filled in by linear interpolation between the closest non-blank fields above and below the blank fields.

#### 13.2.3 Update

The *Update* button redraws the xy series curve in the plot window using the current values in the edit fields.

#### 13.2.4 Insert

The *Insert* button adds a new point to the xy series list by adding a pair of blank fields just above the field containing the cursor.

# 13.2.5 Compress

The *Compress* button reduces the length of the xy series by removing all points whose edit fields are blank.

## 13.2.6 XY Options

The XY Options button brings up the XY Series dialog shown in Figure 13.2. The top group of controls, labeled "time" in the figure is provided for manipulation of the x series. The lower group labeled "vx" is provided for the y series. The titles such as "time" and "vx" change based on what is being edited. The groups can be used to generate or replace the values in the x series, the y series, or both. If the check box just below the x title ("time") is selected, a beginning value, and increment, and a percent change can be input for the x range. These values are applied when the OK button is selected. All of the x values are replaced by a new series generated with the specified parameters. Likewise, if the box beneath the y title ("vx") is selected, the values in the y series can be generated or redefined. The number of new values generated is specified at the bottom of the dialog. If the check boxes by the titles are not selected, the xy values are unaltered when the OK button is selected to exit the dialog.

The XY Options dialog can also be used to define whether the x or y series values should be interpreted as either absolute or relative (delta). If the delta option is chosen, the values beyond the initial value are interpreted as offsets from the previous value. For example, the x series (0.0, 1.0, 1.0, 1.5, 1.5) would actually represent the values (0.0, 1.0, 2.0, 3.5, 5.0).

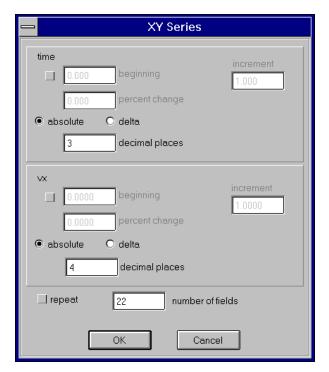


Figure 13.2 The XY Series Dialog.

The *decimal places* field controls how many decimal places are used to display the numbers in the edit fields of the *XY Series Editor*.

The *repeat* toggle is used to define a cyclic series. If the *repeat* toggle is selected, the series is assumed to repeat indefinitely. This information is saved to a file when the series is exported.

#### 13.3 The XY Series Plot

The window in the upper right hand corner of the *XY Series Editor* is used to plot the series corresponding to the xy values in the edit fields. As each value in the edit fields is edited, the corresponding point on the series is adjusted instantaneously. The plot provides an immediate visual feedback to the user, which is helpful in detecting erroneous input values.

#### 13.3.1 The Plot Tools

The *Plot Window* can also be used to edit the xy series graphically. The following tools (found on the right side of the plot) are used for graphical editing:

# The Select Point Tool

The Select Point tool is used to graphically change the xy values of a point by clicking then dragging it to a new location. The tool can also be used to select points for deletion. A set of points can be selected by clicking on the points with the SHIFT key depressed or by dragging a box around a set of points. The selected points can then be deleted by selecting the Delete button beneath the xy edit fields.



### The Create Point Tool

The Create Point tool is used to add new points to a series by clicking in the plot window at the location of the new point.



#### The Zoom Tool

The Zoom tool is used to zoom in on a region of the current series. Clicking on a point zooms the view by a factor of two around the point. Dragging a rectangle alters the mapping so that the region in the rectangle fills the plot window. Holding the SHIFT key down while clicking in the plot window causes the view to be enlarged by a factor of two around the point clicked.

#### 13.3.2 The Plot Macros

The buttons to the lower left of the plot window in the XY Series Editor are used to pan the view in plot window up, down, left, or right. After altering the view using either the Pan button or the Zoom tool, the series can be centered in the plot window by selecting the *Frame* button beneath the plot window.

The buttons to the upper right of the plot window are used to quickly create a series using analytic functions. Each button brings up a dialog that allows the parameters of a function (e.g., sine curve) to be specified from which a series is created.

CHAPTER 14

# File Formats

This chapter contains the file formats for most of the files used by *SMS*. Model specific files such as those used by *FESWMS* and *RMA2* are documented in their respective reference documentation.

Some of the files used by *SMS* use a modified form of the *HEC* style card type format. With this format, the different components of the file are grouped into logical groups called "cards." The first component of each card is a short name which serves as the card identifier. The remaining fields on the line contain the information associated with the card. In some cases, such as lists, a card can use multiple lines.

While card style input makes the file slightly more verbose, there are many advantages associated with the card type approach to formatting files. Some of the advantages are:

- 1. Card identifiers make the file easier to read. Each input line has a label which helps to identify the data on the line.
- 2. The cards names are useful as text strings for searching in a large file. All input lines of a particular type can be located quickly in a large input file.
- 3. In many cases, Cards allow the data to be input in any order (i.e., the order that the cards appear in the file is usually not important).
- 4. Cards make it easy to modify a file format. New data can be included simply by defining a new card type. If the new card is optional (which is typically the case for new cards) old files are still compatible. If an old card type is no longer used, the card can simply be ignored without causing input errors.

# 14.1 SMS Super Files

A SMS super file is a file which contains a list of other files Each of the files in the list must be one of the basic SMS file types (2D meshes, 2D scatter points, materials, TINs). If a super file is selected using the *Open* command in the *File* menu, each of the files listed in the super file are opened and imported. This makes it possible to quickly read in several files without having to identify each file individually in the file browser.

The file format for a super file is shown in Figure 14.1. The first line in the file is the SUPER card, which identifies the file as a super file. Each of the other cards shown are optional. Each of the file cards has a card identifier representing the type of file. The identifier is followed by a file name. The file name should be a complete path if the file is not in the same directory as the super file. Any suffix may be used for the file name. A sample super file is shown in Figure 14.2.

```
SUPER /* File type identifier */
MAT filename /* Material File */
SCAT2D filename /* 2D scatter point file */
MAP filename /* Map file */
MESH2D filename /* 2D Mesh file */
DATA filename /* Dataset File */
STNGS filename /* Settings (*.ini) File */
IMAGE filename /* Image file */
```

Figure 14.1 Super File Format.

```
SUPER
MAT c:\SMS\DATA\SITE1\site1.mat
SCAT2D c:\SMS\DATA\SITE1\site1.xyf
```

Figure 14.2 Sample Super File.

# 14.2 2D Mesh Files

Two-dimensional finite element meshes can be stored in 2D mesh files. This file is an option for storing the geometric data representing the mesh. The *SMS* element types are discussed in Chapter 4. The file format for 2D meshes is shown in Figure 14.3.

```
MESH2D /* File type identifier */
E3T id n1 n2 n3 mat /* 3 node triangle */
E6T id n1 n2 n3 n4 n5 n6 mat /* 6 node triangle */
E4Q id n1 n2 n3 n4 mat /* 4 node quad */
E8Q id n1 n2 n3 n4 n5 n6 n7 n8 mat /* Three node triangle */
ND id x y z /* Nodal coordinates */
```

Figure 14.3 2D-Mesh File Format.

The card types used in the 2D mesh file are as follows:

Card Type	MESH2D
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	E3T	E3T		
Description	Defines a thre	ee node	(linear) triangular element	
Required	NO			
Format	E3T id n1 n2 n	E3T id n1 n2 n3 mat		
Sample	E3T 283 13 32	E3T 283 13 32 27 4		
Field	Variable	Value	Description	
1	id	+	The ID of the element.	
2-4	n1-n3	+	The nodal indices of the element ordered	
			counterclockwise.	
5	mat	+	The material ID for the element.	

Card Type	E6T		
Description	Defines a six node (quadratic) triangular element.		
Required	NO		
Format	E6T id n1 n2 n	3 n4 n5	n6 mat
Sample	E6T 283 13 32 27 22 25 30 4		
Field	Variable	Value	Description
1	id	+	The ID of the element.
2-7	n1-n6	+	The nodal indices of the element ordered
			counterclockwise starting at a corner node.
8	mat	+	The material ID for the element.

Card Type	E4Q		
Description	Defines a four node (linear) quadrilateral element.		
Required	NO		
Format	E4Q id n1 n2 n	.3 n4 ma	t
Sample	E4Q 283 13 32 27 30 4		
Field	Variable	Value	Description
1	id	+	The ID of the element.
2-5	n1-n4	+	The nodal indices of the element ordered
			counterclockwise.
6	mat	+	The material ID for the element.

Card Type	E8Q	E8Q		
Description	Defines an eig	ht node	(quadratic) quadrilateral element.	
Required	NO			
Format	E8T id n1 n2 n	E8T id n1 n2 n3 n4 n5 n6 n7 n8 mat		
Sample	E8T 283 13 32	E8T 283 13 32 27 22 25 30 29 31 4		
Field	Variable	Variable Value Description		
1	id	+	The ID of the element.	
2-9	n1-n8	+	The nodal indices of the element ordered	
			counterclockwise starting at a corner node.	
10	mat	+	The material ID for the element.	

Card Type	ND	ND			
Description	Defines the co	ordinat	es of a node.		
Required	NO				
Format	ND id x y z	ND id x y z			
Sample	ND 84 120.4 38	ND 84 120.4 380.3 5632.0			
Field	Variable	Value	Description		
1	id	+	The ID of the node.		
2-4	х, у, z	±	The nodal coordinates.		

#### 14.3 2D Scatter Point Files

Two-dimensional scatter point sets are stored in 2D scatter point files. Multiple scatter point sets can be stored in a single file. Each point in a scatter point set is defined by a pair of XY coordinates.

The format of the 2D scatter file is shown in Figure 14.4. There are two types of scatter point cards files: XY and XYD. With the XY card, only the XY coordinates are defined. It is assumed that any data sets that will be associated with the scatter point set will be imported from a data set file after the scatter point file is imported.

With the XYD card, one or more data values may be associated with each scatter point in the file. These values are converted to data sets on input. This approach makes it easy to import a scatter point set with scalar values for each point.

Scatter point sets can be saved to a file using either the XY or XYD formats. (See Chapter 5 for more information on reading and saving scatter point files.)

Figure 14.4 2D Scatter Point File Format.

```
SCAT2D
BEGSET
NAME "lakes"
ID 8493
DELEV 0.000000000000e+00
IXY 25
1 1.470000000000e+02 3.9000000000e+02
2 8.8200000000e+02 9.4900000000e+02
.
.
24 1.730000000000e+02 7.01000000000e+02
ENDSET
```

Figure 14.5 Sample 2D Scatter Point File.

The cards used in the 2D scatter point file are as follows:

Card Type	SCAT2D
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	BEGSET
Description	Identifies the beginning of a scatter point set. No fields.
Required	NO

Card Type	NAME			
Description	Defines the name	for the fol	llowing scatter point set.	
Required	NO			
Format	NAME "name"	NAME "name"		
Sample	NAME "st mary	NAME "st mary"		
Field	Variable Value Description			
1	name	str	The name for the following scatter points. Remains	
			as default until new NAME card is encountered.	

Card Type	ID			
Description	Defines the ID for	Defines the ID for the scatter point set.		
Required	YES			
Format	ID id	ID id		
Sample	ID 43098			
Field	Variable	Value	Description	
1	id	+	The ID for the following scatter point set.	

Card Type	DELEV			
Description	Defines the defau	ılt elevatio	n for the scatter point set.	
Required	NO			
Format	DELEV el	DELEV el		
Sample	DELEV 9.0	DELEV 9.0		
Field	Variable	Variable Value Description		
1	el	±	The default elevation for the following scatter points.	
			Remains as default until new DELEV card is	
			encountered.	

Card Type	IXY			
Description	Defines a scatter	Defines a scatter point set.		
Required	YES			
Format	IXY np id <sub>1</sub> x <sub>1</sub> y <sub>1</sub> id <sub>2</sub> x <sub>2</sub> y <sub>2</sub> id <sub>np</sub> x <sub>np</sub> y <sub>np</sub>			
Sample	IXY 4 1 12.3 34.5 2 52.2 23.5 3 63.2 27.4 4 91.1 29.3			
Field	Variable	Value	Description	
1	np	+	The number of scatter points in the scatter point set.	
2	id	+	The ids of the points.	
3-4	х,у	±	The coordinates of the points.	
Repeat fields 2-4 np times		_		

Card Type	ENDSET
Description	Identifies the end of a scatter point set. No fields.
Required	NO

#### 14.4 ASCII Data Set Files

Data sets can be stored to either ASCII or binary files. Multiple data sets can be stored in a single file and both scalar and vector data sets can be saved to the same file. The ASCII data set format is shown in Figure 14.6. A sample data set file is shown in Figure 14.7.

For scalar data set files, one value is listed per vertex, cell, node, or scatter point. For vector data set files, one set of XYZ vector components is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

If variograms have been defined for a data set or time steps of a data set within SMS, the variograms are saved in the data set file. The variogram cards are not documented.

```
DATASET
                  /* File type identifier */
                  /* Type of object data set is associated with */
OBJTYPE type
                  /* Beginning of scalar data set */
BEGSCL
                 /* Object id */
OBJID id
                  /* Number of data values */
ND numdata
                 /* Number of cells or elements */
NC numcells
                 /* Data set name */
NAME "name"
TS istat time
                /* Time step of the following data. */
stat1
                  /* Status flags */
stat2
{\tt stat}_{\tt numcells}
                  /* Scalar data values */
val<sub>1</sub>
val2
valnumdata
/* Repeat TS card for each time step */
ENDDS /* End of data set */
                  /* Beginning of vector dataset */
BEGVEC
VECTYPE type
                 /* Vector at node/gridnode or element/cell */
                 /* Object id */
OBJID id
                 /* Number of data values */
ND numdata
                 /* Number of cells or elements */
NC numcells
                 /* Data set name */
NAME "name"
                 /* Time step of the following data. */
TS istat time
                  /* Status flags */
stat1
stat2
stat_{numcells}
vx1 vy1
v_{x2} v_{y2}
vnumdata vnumdata vnumdata
/* Repeat TS card for each time step */
                 /* End of data set */
^{\prime} Repeat BEGSCL and BEGVEC sequences for each data set ^{*}/
```

Figure 14.6 ASCII Data Set File Format.

```
DATASET
OBJTYPE grid2d
BEGSCL
OBJID 27211
ND 8
NC 8
NAME "sediment transport"
TS 1 1.00000000e+00
0
0
1
1
0.00000000e+00
0.00000000e+00
0.00000000e+00
3.24000000e+00
4.39000000e+00
2.96000000e+00
7.48000000e+00
0.00000000e+00
ENDDS
BEGVEC
VECTYPE 0
OBJID 27211
ND 8
NC 8
NAME "velocity"
TS 1 5.00000000e+00
0
1
1
1
0
1.60000000e+01 1.60000000e+01
1.44000000e+01 6.40000000e+01
1.44000000e+02 1.44000000e+02
1.96000000e+02 1.96000000e+02
2.25000000e+02 2.25000000e+02
9.21600000e+03 9.21600000e+03
9.60400000e+03 9.60400000e+03 9.80100000e+03 9.80100000e+03
```

Figure 14.7 Sample ASCII Data Set File.

The card types used in the scalar data set file format are as follows:

Card Type	DATASET
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	OBJTYPE			
Description	Identifies the type of objects that the data sets in the file are associated with.			
Required	YES. If card does not exist, the file can only be read through the Data Browser. The data sets would then be assigned to the objects corresponding to the active module.			
Format	OBJTYPE type			
Sample	OBJTYPE tin			
Field	Variable	Value	Description	
1	type	tin	TINs	
		mesh2d	2D meshes	
		scat2d	2D scatter points	

Card Type	BEGSCL
Description	Scalar data set file identifier. Marks beginning of scalar data set. No fields.
Required	YES

Card Type	BEGVEC		
Description	Vector data set file identifier. Marks beginning of vector data set. No fields.		
Required	YES		

Card Type	VECTYPE						
Card ID	150	150					
Description	Identifies the	Identifies the type of vector data that will be read and where to apply it.					
Required	this card is r	This card is only required if the vector data is associated with elements/cells. If this card is not present, it is assumed that the data are associated with nodes/gridnodes.					
Field	Variable	Variable Size Value Description					
1	type	4 byte int	0	The vectors will be applied to the nodes/gridnodes.			
			1	The vectors will be applied to the elements/cells.			

Card Type	ND			
Description	The number of data values that will be listed per time step. This number should correspond to the total number of vertices, nodes, cells centers (cell-centered grid), cell corners (mesh-centered grid), maximum node id (meshes) or scatter points.			
Required	YES	YES		
Format	ND numdata	ND numdata		
Sample	ND 10098	ND 10098		
Field	Variable Value Description			
1	numdata	+	The number of items. At each time step, numdata values are printed.	

Card Type	NC			
Description	This number should correspond to the maximum element id (meshes) or the			
	number of cells (g	number of cells (grids).		
Required	YES	YES		
Format	NC numcells			
Sample	NC 3982			
Field	Variable Value Description			
1	numcells	+	The number of elements or cells.	

Card Type	NAME			
Description	The name of the d	The name of the data set.		
Required	YES	YES		
Format	NAME "name"			
Sample	NAME "Total he	NAME "Total head"		
Field	Variable Value Description			
1	"name"	str	The name of the dataset in double quotes.	

Card Type	TS				
Description	Marks the beginning of a new time step, indicates if stat flags are given, and				
	defines the time step value, status flags, and scalar data values for each item.				
Required	YES				
Format	TS istat time stat1 stat2 stat numcells val1 val2 valnumdata				
Sample	TS 1 12.5 0 1 1 1 34.5 74.3 58.4 72.9				
Field	Variable	Value	Description		
1	istat	0	Use status flags from previous time step. For first time step, this indicates that all cells are active. Status flags will be listed.		
2	time	+	The time step value. If only one time step exists, time is not required		
2 - (n+1)	stat	0,1	The status of each item. If active, stat=1. If inactive stat=0. Omitted if i=0 on STAT card.		
(n+2) -	val	±	The scalar data values of each item.		

# 14.5 Binary Data Set Files

Data sets can be stored to either ASCII or binary files. Compared to ASCII files, binary files require less memory and can be imported to SMS more quickly. The disadvantages of binary files are that they are not as portable and they cannot be viewed with a text editor.

The binary data set file format is shown in Figure 14.11. The binary format is patterned after the ASCII format in that the data are grouped into "cards". However, the cards are identified by a number rather than a card title.

Item	Size	Description
version	4 byte integer	The SMS binary data set file format version. value = 3000.
objecttype	4 byte integer	Identifies the type of objects that the data sets in the file are associated with. Options are as follows:  1 TINs 3 2D meshes 5 2D scatter points
SFLT	4 byte integer	The number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).
SFLG	4 byte integer	The number of bytes that will be used in the remainder of the file for status flags.
BEGSCL or BEGVEC		Marks the beginning of a set of cards defining a scalar or vector data set.
VECTYPE	4 byte integer	(0 or 1) In the case of vector data set files, indicates whether the vectors will be applied at the nodes/gridnodes or the elements/cells.
OBJID	4 byte integer	The id of the associated object. Value is ignored for grids and meshes.
NUMDATA	4 byte integer	The number of data values that will be listed per time step. This number should correspond to the number of vertices, nodes, cell centers (cell-centered grid), cell corners (mesh-centered grid) or scatter points.
NUMCELLS	4 byte integer	This number should correspond to the number of elements (meshes) or the number of cells (mesh-centered grids). Value is ignored for other object types.
NAME	40 bytes	The name of the dataset. Use one character per byte. Mark the end of the string with the '\0' character.
TS		Marks the beginning of a time step.
ISTAT	SFLG integer	(0 or 1) Indicates whether or not status flags will be included in the file.
TIME	SFLT real	Time corresponding to the time step.
statflag1	SFLG integer	Status flag (0 or 1) for node 1
statflag2	SFLG integer	Status flag (0 or 1) for node 2
val1	SFLT real	Scalar value for item 1
		Scalar value for item 2
	C. Li ioui	Could value for nom 2
		Repeat card 200 for each timestep in the data set.
ENDDS	1	Signal the end of a set of cards defining a data set.
	version objecttype  SFLT  SFLG  BEGSCL or BEGVEC VECTYPE  OBJID  NUMDATA  NUMCELLS  NAME  TS ISTAT  TIME statflag1	version 4 byte integer  objecttype 4 byte integer  SFLT 4 byte integer  SFLG 4 byte integer  BEGSCL or BEGVEC  VECTYPE 4 byte integer  OBJID 4 byte integer  NUMDATA 4 byte integer  NUMCELLS 4 byte integer  NAME 40 bytes  TS  ISTAT SFLG integer  TIME SFLT real statflag1 SFLG integer  statflag2 SFLG integer  val1 SFLT real val2 SFLT real

Figure 14.8 The Binary Scalar or Vector Data Set File Format.

The cards in the binary data set file are as follows:

Card Type	VERSION
Card ID	3000
Description	File type identifier. No fields.
Required	YES

Card Type	OBJTYPE			
Card ID	100			
Description	Identifies the	e type of obje	cts that the d	ata sets in the file are associated with.
Required			,	n only be read through the Data Browser.  I to the objects corresponding to the active
Field	Variable	Size	Value	Description
1	id	4 byte int	1	TINs
			3	2D meshes
			5	2D scatter points

Card Type	SFLT				
Card ID	110				
Description		Identifies the number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).			
Required	YES	YES			
Field	Variable	Size	Value	Description	
1	sizefloat	4 byte int	4, 8, or 16	Number of bytes	

Card Type	SFLG				
Card ID	120				
Description	Identifies the	Identifies the number of bytes that will be used in the remainder of the file for			
	status flags	(1, 2, or 4).			
Required	YES				
Field	Variable	Size	Value	Description	
1	sizeflag	4 byte int	1, 2, or 4	Number of bytes	

Card Type	BEGSCL
Card ID	130
Description	Marks the beginning of a set of cards defining a scalar data set.
Required	YES

Card Type	BEGVEC
Card ID	140
Description	Marks the beginning of a set of cards defining a vector data set.
Required	YES

Card Type	VECTYPE				
Card ID	150				
Description	Identifies the	e type of vect	or data that w	vill be read and where to apply it.	
Required		not present, it		data is associated with elements/cells. If that the data are associated with	
Field	Variable	Size	Value	Description	
1	type	4 byte int	0	The vectors will be applied to the nodes/gridnodes.	
			1	The vectors will be applied to the elements/cells.	

Card Type	OBJID				
Card ID	160				
Description	The id of the	The id of the associated object.			
Required	This card is	This card is required in the case of TINs, 2D scatter points, and 3D scatter points.			
	With each of	With each of these objects, multiple objects may be defined at once. Hence the			
	id is necessary to relate the data set to the proper object.				
Field	Variable	Size	Value	Description	
1	id	4 byte int	+	The id of the object.	

Card Type	NUMDATA	NUMDATA			
Card ID	170				
Description	correspond	The number of data values that will be listed per time step. This number should correspond to the number of vertices, nodes, cell centers (cell-centered grid), cell corners (mesh-centered grid), maximum node id (meshes) or scatter points.			
Required	YES				
Field	Variable	Size	Value	Description	
1	numdata	4 byte int	+	The number of items. At each timestep, numdata are listed.	

Card Type	NUMCELLS	6		
Card ID	180			
Description	This number (grids).	r should corre	espond to the	element id (meshes) or the number of cells
Required	YES			
Field	Variable	Size	Value	Description
1	numcells	4 byte int	+	The number of elements or cells.

Card Type	NAME			
Card ID	190			
Description	The name of	f the data set		
Required	YES			
Field	Variable	Size	Value	Description
1	name	40 bytes	str	The name of the data set. Use one character per byte. Mark the end of the string with the '\0' character.

Card Type	TS						
Card ID	200	200					
Description		Defines the set of scalar values associated with a timestep. Should be repeated for each time step.					
Required	YES						
Field	Variable	Size	Value	Description			
1	istat	SFLG int	0	Use status flags from previous time step. For the first time step, this value indicates that all cells are active. Status flags will be listed.			
2	time	SFLT int	+	The time step value. This number is ignored if ther is only one time step.			
	stat	SFLG int	0	Inactive Active One status flag should be listed for each cell or element. These flags are included only when istat = 1.			
	val	SFLT real	±	The scalar values			

Card Type	ENDDS
Card ID	210
Description	Signals the end of a set of cards defining a data set
Required	YES

# 14.6 ASCII Scalar Data Set Files (version 4)

Scalar data sets can be stored to either ASCII or binary files. The ASCII data set format is shown in Figure 14.9. One scalar value is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

```
SCALAR /* File type identifier */
ND n /* Number of data values */
STAT i /* Status flags will be included in file */
TS time /* Time step of the following data. */
stat1 /* Status flags */
stat2

.
.
statn
val1 /* Scalar data values */
val2
.
.
valn
/* Repeat TS card for each time step */
```

Figure 14.9 ASCII Scalar Data Set File Format.

The card types used in the scalar data set file format are as follows:

Card Type	SCALAR
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	ND			
Description	should corresp	Defines the number of data values per time step. This number should correspond to the total number of vertices, nodes, cells, or scatter points in the object the data set is imported to.		
Required	YES	YES		
Format	ND n	ND n		
Sample	ND 4			
Field	Variable	Variable Value Description		
1	n	+	The number of data values per time step.	

Card Type	STAT			
Description			not status flags will be included in the ms are assumed to be active.	
	TITE. II HOU	, all ite	ils are assumed to be active.	
Required	YES			
Format	STAT i	STAT i		
Sample	STAT 0			
Field	Variable	Value	Description	
1	i	0,1	If status flags are to be included in the file, i=1. If status flags are not to be included in file, i=0.	

Card Type	TS		
Description	Defines a set	of scala	ar values associated with a time step.
Required	YES		
Format	TS time stat1 stat2 . stat $_{\rm n}$ val1 val2 . val $_{\rm n}$		
Sample	TS 12.5 0 1 1 1 34.5 74.3 48.3 72.9		
Field	Variable	Value	Description
1	time	±	The time step value. Ignored if there is only one time step.
2 - (n+1)	stat	0,1	The item status. If active, stat=1. If inactive stat=0. Omitted if i=0 on STAT card.
(n+2) - (2n+1)	val	±	The scalar data values of each item.

# 14.7 ASCII Vector Data Set Files (version 4)

Vector data sets can be stored to either ASCII or binary files. The ASCII data set format is shown in Figure 14.10. One set of xyz vector components is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

```
VECTOR /* File type identifier */
ND n /* Number of data values */
STAT i /* Status flags will be included in file */
TS time /* Time step of the following data. */
stat1 /* Status flags */
stat2
.
.
. statn
Vx1 Vy1 Vz1 /* Vector data values */
Vx2 Vy2 Vz2
.
.
. vxn Vyn Vzn
/* Repeat TS card for each time step */
```

Figure 14.10 ASCII Vector Data Set File Format.

The card types used in the vector data set file format are as follows:

Card Type	VECTOR
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	ND		
Description	should corresp	ond to t	f data values per time step. This number the total number of vertices, nodes, cells, the object the data set is imported to.
Required	YES		
Format	ND n		
Sample	ND 4		
Field	Variable	Value	Description
1	n	+	The number of data values per time step.

Card Type	STAT		
Description	Specified whether or not status flags will be included in the file. If not, all items are assumed to be active.		
Required	YES		
Format	STAT i		
Sample	STAT 0		
Field	Variable	Value	Description
1	i	0,1	If status flags are to be included in the file, i=1. If status flags are not to be included in file, i=0.

Card Type	TS		
Description	Defines a set	of vecto	or values associated with a time step.
Required	YES		
Format	TS time stat1 stat2 . statn vx1 vy1 vz1 vx2 vy2 vz2 . vxn vyn vzn		
Sample	TS 12.5 0 1 1 1 3.4 -5.6 0.0 6.2 -8.9 0.0 -2.9 6.2 0.0 -1.2 -3.4 0.0		
Field	Variable	Value	Description
1	time	±	The time step value. Ignored if there is only one time step.
2 - (n+1)	stat	0,1	The item status. If active, stat=1. If inactive stat=0. Omitted if i=0 on STAT card.
(n+2) - (4n+1)	$v_x, v_y, v_z$	±	The xyz components of each vector.

# 14.8 Binary Scalar Data Set Files (version 4)

Scalar data sets can be stored to either ASCII or binary files. The binary data set file format is shown in Figure 14.11. One scalar value is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

Item	Size	Description
version	4 byte integer	Version = 1000 for scalar file
n	4 byte integer	Number of items (cells, nodes, etc.)
status data	4 byte integer	(0 or 1) Indicates whether or not status flags will be included in the file.
SFLT	4 byte integer	The number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).
SFLG	4 byte integer	The number of bytes that will be used in the remainder of the file for each status flag (1, 2, or 4).
time step I	SFLT real	Time corresponding to time step I
statflag1	SFLG integer	Status flag (0 or 1) for node 1
statflag2	SFLG integer	Status flag (0 or 1) for node 2
statflag3	SFLG integer	Status flag (0 or 1) for node 3
val1	SFLT real	Scalar value for item 1
val2	SFLT real	Scalar value for item 2
val3	SFLT real	Scalar value for item 3
Repeat time step		
group for each time		
step.		

Figure 14.11 The Binary Scalar Data Set File Format.

The following sample code illustrates how binary scalar files are written using FORTRAN:

Explanation of variables:

Variable	Description	
IVERSION	Version = 1000 for scalar file	
NNP	Number of items (cells, nodes, etc.)	
ISTAT	(0 or 1) Indicates whether or not status flags will be included in the file.	
ISFLT	The number of bytes that will be used in the remainder of the file for each floating point value. (4, 8 or 16)	
ISFLG	The number of bytes that will be used in the remainder of the file for each status flag. (1, 2 or 4)	
NTS	The number of time steps (separate sets of data corresponding to the	
	vertex, cell, node, or scatter point group)	
TIME(I)	Time corresponding to time step I	
IFLG(I,J)	Status flag (0 or 1) for node J of time step I	
VALUE(I,J)	Scalar value for item J of time step I	

**Note**: Most FORTRAN compilers use the convention of writing four byte block size markers before and after each set of data written to a binary (UNFORMATTED) file. Block size markers indicate the size of the data that comes next in the file defined by individual WRITE statements. *SMS* expects that the data has been written to the binary file with the same configuration of WRITE statements shown in the previous example. *SMS* can also read a binary file without block size markers (standard C convention).

# 14.9 Binary Vector Data Set Files (version 4)

Vector data sets can be stored to either ASCII or binary files. The binary data set file format is shown in Figure 14.12. One set of vector values is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

Item	Size	Description
version	4 byte integer	Version = 2000 for vector file.
n	4 byte integer	Number of items (nodes, cells, etc.)
status data	4 byte integer	(0 or 1) Indicates whether or not status flags will be included in the file.
SFLT	4 byte integer	The number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).
SFLG	4 byte integer	The number of bytes that will be used in the remainder of the file for each status flag (1, 2, or 4).
time step I	SFLT real	Time corresponding to time step I
statflag1	SFLG integer	Status flag (0 or 1) for item 1
statflag2	SFLG integer	Status flag (0 or 1) for item 2
statflag3	SFLG integer	Status flag (0 or 1) for item 3
vx1	SFLT real	X comp. of vector. for item 1
vy1	SFLT real	Y comp. of vector for item 1
vz1	SFLT real	Z comp. of vector for item 1
Repeat time step		
group for each time		
step.		

Figure 14.12 The Binary Vector Data Set File Format.

The following sample code illustrates how binary vector files are written using FORTRAN:

```
OPEN(30, FILE='vector.bin', FORM='UNFORMATTED', STATUS='UNKNOWN')
WRITE(30) IVERSION, NNP, ISTAT, ISFLT, ISFLG
DO 10 !=1, NTS
    WRITE(30) TIME(I)
    IF(ISTAT .NE. 0)THEN
        WRITE(30)(IFLG(I,J), J=1, NNP)
    END IF
    WRITE(30) (X(I,J), Y(I,J), Z(I,J), J=1, NNP)
10 CONTINUE
```

Explanation of variables

Variable	Description		
IVERSION	Version = 2000 for vector file		
NNP	Number of items (cells, nodes, etc.)		
ISTAT	(0 or 1) Indicates whether or not status flags will be included in the file.		
ISFLT	The number of bytes that will be used in the remainder of the file for each floating point value. (4, 8 or 16)		
ISFLG	The number of bytes that will be used in the remainder of the file for each status flag. (1, 2 or 4)		
NTS	The number of time steps (separate sets of data corresponding to the vertex, cell, node, or scatter point group)		
TIME(I)	Time corresponding to time step I		
IFLG(I,J)	Status flag (0 or 1) for node J of time step I		
X(I,J)	X comp. of vector. for item J of time step I		
Y(I,J)	Y comp. of vector. for item J of time step I		
Z(I,J)	Z comp. of vector. for item J of time step I		

**Note**: Most FORTRAN compilers use the convention of writing four byte block size markers before and after each set of data written to a binary (UNFORMATTED) file. Block size markers indicate the size of the data that comes next in the file defined by individual WRITE statements. *SMS* expects that the data has been written to the binary file with the same configuration of WRITE statements shown in the example. *SMS* will also read a binary file without block size markers (standard C convention).

# 14.10 Gage Files

Gages are used to generate curves representing the variation of a dynamic data set with time at particular location in a grid or mesh. Gage plots are useful in the process of calibrating a model to measured field data. Gage files are used to import a set of gages and a set of measured curves at each gage. This allows a set of measured curves to be compared to a set of computed curves.

The format of the gage file is shown in Figure 14.13. Several gages can be included in a single file. In addition, several vector or scalar curves can be associated with each gage. The times for each of the curves do not need to match.

```
/* File type identifier */
BEGGAG
       /* The beginning of a gage group. */
NAM name
                         /* The name of the gage. */
                         /* The gage location.
LOC x y z
                         /* The positive direction vector. */
DIR vx vy vz
COL red green blue /* The color of the gage. */
SCA np name /* A scalar curve. */
t1 v1 /* Time and value, from 1 to np. */
t2 v2
tnp vnp
VEC np name
                         /* A vector curve. */
                         /* Time vector, from 1 to np. */
t1 vx1 vy1 vz1
t2 vx2 vy2 vz2
tnp vxnp vynp vznp
/* Repeat scalar and vector cards as many times as necessary. */
ENDGAG /* The end of a gage group */
 Repeat gage card group as many times as necessary.
```

Figure 14.13 Gage File Format.

The card types used in the gage file format are as follows:

Card Type	GAGE
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	BEGGAG
Description	Marks the beginning of a group of cards describing a gage. There should be a corresponding ENDGAG card at a latter point in the file. No fields.
Required	YES

Card Type	NAM			
Description	Defines a name	to be a	associated with the gage.	
Required	NO			
Format	NAM name	NAM name		
Sample	NAM Gage #23GP	NAM Gage #23GP		
Field	Variable	Value	Description	
1	name	str	The name of the gage	

Card Type	LOC	LOC			
Description	Defines the xy	Defines the xyz coordinates of the gage.			
Required	YES	YES			
Format	LOC x y z	LOC x y z			
Sample	LOC 348.23 623	LOC 348.23 623.36 93.2			
Field	Variable	Value	Description		
1-3	х, у, z	±	The coordinates of the gage.		

Card Type	DIR			
Description	Defines the po (+ or -)) for		direction vector(used to determine the sign e.	
Required	NO			
Format	DIR vx vy vz	DIR vx vy vz		
Sample	DIR 0.45 0.55	DIR 0.45 0.55 0.0		
Field	Variable	Value	Description	
1-3	vx, vy, vz	±	The components of the unnormalized positive direction vector.	

Card Type	COL			
Description	Identifies a color to be associated with the gage.			
Required	NO			
Format	COL id red green blue			
Sample	COL 124 67 245			
Field	Variable	Value	Description	
1-3	red, green,	0-255	The values of the red, green, and blue components of	
	blue		the color.	

Card Type	SCA			
Description	Defines a curve of scalar values			
Required	NO			
Format	SCA np name t1 v1 t2 v2 . tnp vnp			
Sample	SCA 6 head 0.0 100.0 0.5 99.93 1.0 99.14 1.5 98.64 2.0 98.32 2.5 98.12			
Field	Variable	Value	Description	
1	np	+	The number of points in the curve.	
2	name	str	The name of the curve. If curves of the same type are to be used with other gages (i.e. head), the names should be spelled exactly the same on the SCA cards for the other gages. This will ensure that the curves are treated as a single group in the Curves dialog.	
3	t	±	The time value for a point on the curve.	
4	V	±	The scalar value for a point on the curve.	
Fields 3-4 should be repeated np times.				

Card Type	VEC			
Description	Defines a curve of vector values			
Required	NO			
Format	VEC np name			
	t1 vx1 vy1 vz1			
	t2 vx2 vy2 vz2			
	•			
		****		
	tnp vxnp vynp			
Sample	SCA 6 velocity 0.0 0.018 0.02			
	0.5 0.033 0.02			
	1.0 0.035 0.028 0.0			
	1.5 0.032 0.022 0.0			
	2.0 0.023 0.01 2.5 0.015 0.00			
Field	Variable	Value	Description	
1	np	+	The number of points in the curve.	
2	name	str	The name of the curve. If curves of the same type are	
			to be used with other gages, the names should be	
			spelled exactly the same on the SCA cards for the	
			other gages. This will ensure that the curves are	
			treated as a single group in the Curves dialog.	
3	t	±	The time value for a point on the curve.	
4-6	vx, vy, vz	±	The vector components for a point on the curve.	
Fields 3-6				
should be				
repeated np				
times.				

Card Type	ENDGAG
Description	Marks the end of a group of cards describing a gage. There should be a corresponding BEGGAG card at a previous point in the file. No fields.
Required	YES

## 14.11 Map Files

Map files are used to store feature object and drawing object data. Feature objects include points, nodes, vertices, arcs and polygons. Drawing objects include rectangles, ovals, lines, and text. The map file also includes the grid frame. The map file format is shown in Figure 14.14. Figure 14.14 does not include the cards defining feature object attributes. These cards are described below. A sample map file is shown in Figure 14.15.

```
MAP
                                 /* file type identifier */
BEGCOV
                                 /* beginning of a new coverage */
COVNAME "name"
                                 /* coverage name */
                                 /* coverage attributes type */
COVATTS attstype
                                 /* new point identifier
POINT
                                 /* xy coordinates of the point*/
хү х у
ID id
                                 /* id of the point*/
                                /* end of data for point*/
END
                                 /* new node identifier
NODE
                                 /* xy coordinates of node*/
хү х у
ID id
                                 /* id of node *
END
                                /* end of data for node */
                                /* new arc identifier */
ARC
                                /* id of arc */
                                /* ids of beginning and ending nodes for arc */
NODES id1 id2
                                /* number of vertices between nodes for arc */
ARCVERTICES n
                                /* xy location of vertices */
x1 y1
x2 y2
xn yn
ARCBIAS Value
                                 /* bias value for meshing */
END
                                 /* end of data for arc 1
                                 /* new polygon identifier */
POLYGON
                                 /* id of polygon*/
/* number of boundary arcs for polygon */
ID id
ARCS n
                                 /* ids of boundary arcs */
id1
id2
HARCS n /* number of hole-arcs for polygon */
        /* ids of hole-arcs */
id2
idn
        /* end of data for polygon */
ENDCOV /* end of coverage data
        /* new rectangle identifier */
RECT
C1 x1 y1 z1
C2 x2 y2 z2
                                /* xyz coordinates of corner 1 */    /* xyz coordinates of corner 2 */
C3 x3 y3 z3
C4 x4 y4 z4
THICK width
                                /* xyz coordinates of corner 3 */
                                /* xyz coordinates of corner 4 */
                                 /* line thickness of rectangle border */
STYLE style
                                 /* line style of rectangle */
                                 /* red green blue components of line color */
LINECOL r g b
FILLCOL r g b
                                 /* red green blue components of fill color */
                                 /* fill pattern of rectangle */
FILLPAT pattern
                                 /* Viewing angle rect. was created in */
THETA theta
ALPHA alpha
                                 /* Viewing angle rect. was created in */
END
       /* end of rectangle data */
        /* new oval identifier */
OVAL
                                 /* same cards as rectangle */
END
        /* end of rectangle data */
        /* new line identifier */
VERTS n /* number of points in the line */
                                 /* xyz coordinates of point 1 */
```

```
PT x2 y2 z2
                                      /* xyz coordinates of point 2 */
PT xn yn zn
THICK width
                                      /* xyz coordinates of point n */
/* line thickness */
STYLE style
                                      /* line style */
                                      /* red green blue components of line color */
LINECOL r g b
                                      /* arrow head flag */
ARRHED type
                                      /* arrow head base width */
HEDWID width
                                      /* arrow head length */
HEDlEN length
END /* end of line data */
TEXT /* new text string identifier */
                                     /* text string */
/* xyz location of text */
STRING "str"
LOCAL x y z PCFONT "name"
                                     /* pc font name */
/* unix font name */
UNIXFONT "name"
COLOR r g b /* red END /* end of text string data */
                                      /* red green blue components of text */
GRIDFRAME x y z dx dy dz \theta
                                      /* dimensions of gridframe */
```

Figure 14.14. Map File Format.

```
MAP
LEND
BEGCOV
COVNAME "default coverage"
COVATTS 2DMESH
NODE
XY 174.112149532710280 483.364485981308410
GWNCARD 0.000 0.000 0.000 0.000
END
ARC
ID 1
NODES
ARCVERTICES 2
132.99065 133.83178
630.56075 429.90654
ARCBIAS 1.000000
END
POLYGON
ARCS 0
PATCHPTS 0 0 0 0
HARCS 1
ID 0
ADAPTESS 1.000000
END
POLYGON
ARCS 1
PATCHPTS 0 0 0 0
ADAPTESS 1.000000
CEILING 1 0.577 0.577 0.577
MAT
MN
                               mat 1
MC
       1
                               255
                                              204
MS
       1
END
ENDCOV
```

Figure 14.15 Sample Map File.

The format of the cards in the map file are given below:

#### 14.11.1General

The following card is used to define general map data.

Card Type	MAP
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

## 14.11.2Feature Objects

The following cards are used to define coverages, polygons, arcs, and points.

Card Type	BEGCOV
Description	Beginning of a series of cards definining a coverage.
Required	YES

Card Type	COVNAME			
Description	Coverage name.			
Required	NO	NO		
Format	COVNAME "name	COVNAME "name"		
Sample	COVNAME "gene:	COVNAME "general"		
Field	Variable	Value	Description	
1	name	str	Coverage name.	

Card Type	COVATTS					
Description	Coverage model a	Coverage model attributes set.				
Required	YES	YES				
Format	COVATTS type					
Sample	COVATTS 2DMESH					
Field	Variable	Variable Value Description				
1	type	2DGRID	2D grid.			
		2DMESH	2D mesh			

Card Type	ENDCOV
Description	End of cards defining a coverage.
Required	YES

Card Type	POINT
Description	Beginning of a series of cards defining a point.
Required	NO

Card Type	XY			
Description	xy coordinates of a	xy coordinates of a point or node.		
Required	YES	YES		
Format	ХҮ х у			
Sample	XY 10.0 20.0			
Field	Variable Value Description			
1-2	х, у	±	xy coordinates of the point or node.	

Card Type	ID			
Description	id of a feature obje	id of a feature object.		
Required	YES	YES		
Format	ID id	ID id		
Sample	ID 10	ID 10		
Field	Variable	Value	Description	
1	id	+	id of the feature object	

Card Type	GWN			
Description	1-D element parar	1-D element paramaters		
Required	NO			
Format	GWNCARD Wo Ws	Sl Sr		
Sample	GWNCARD 1.000 2.000 3.000 4.000			
Field	Variable	Value	Description	
1-4	Wo, Ws, S <sub>L</sub> , S <sub>R</sub>	±	1-D element paramaters	

Card Type	END
Description	End of a series of cards defining a feature object or drawing object.
Required	YES

Card Type	NODE
Description	Beginning of a set of cards defining a node.
Required	NO

Card Type	ARC
Description	Beginning of a set of cards defining an arc.
Required	NO

Card Type	NODES			
Description	Beginning and end	Beginning and ending nodes of arc.		
Required	YES			
Format	NODES n1 n2	NODES n1 n2		
Sample	NODES 10 15			
Field	Variable	Value	Description	
1-2	n1, n2	+	id of beginning and ending nodes of an arc.	

Card Type	ARCVERTICES	ARCVERTICES		
Description	Vertices between	Vertices between nodes of an arc identifier.		
Required	Only if the arc has	vertices		
Format	ARCVERTICES nv	vert		
	х1 У1			
	х2 у2			
	•			
	$x_n y_n$			
Sample	ARCVERTICES 3			
'	5.0 10.0			
	12.0 8.0			
	2.5 7.6			
Field	Variable	Value	Description	
1	nvert	+	Number of intermediate vertices	
2-3	х, у	±	Vertex coordinates. Fields 2-3 repeated for each	
			vertex	

Card Type	POLYGON
Description	Polygon Type idenfier.
Required	NO

Card Type	ARCS	ARCS		
Description		Defines the arcs forming the outer boundary of a polygon. There should only be one card of this type for each polygon, except for the universal polygon.		
Required	YES			
Format	ARCS count	ARCS count		
Sample	ARCS 2 10 12			
Field	Variable	Value	Description	
1	count	+	Number of arcs in the polygon.	
2	id	+	Arc id. Field 2 is repeated for each arc in the polygon. The arcs should be listed in clockwise order.	

Card Type	HARCS		
Description	Defines the arcs bounding the holes in the polygon. There should be one card of		
	this type for each	hole in the	e polygon.
Required	NO.		
Format	HARCS count		
Sample	HARCS 2 10 12		
Field	Variable	Value	Description
1	count	+	Number of hole-arcs in the polygon.
2	id	+	Arc id. Field 2 is repeated for each arc in the polygon. The arcs should be listed in counter-clockwise order.

## 14.11.3 Feature Object Attributes

The following cards are used to specify the attributes associated with points, arcs, and polygons. They should be placed in the file between the beginning and ending cards for the associated objects.

#### **Attributes For Points and Nodes:**

Card Type	SPECHHEAD		
Description	Specified Head he	ad attribu	te.
Required	YES if a feature of	oject is of	Specified Head type.
Format	SPECHHEAD con	nstflag	value/id
Sample	SPECHHEAD 1 1:	38.6	
Field	Variable	Value	Description
1	constflag	0,1	The type code:
			0 = Transient attribute represented by xy series.
			1 = Constant attribute represented by a single value.
2	value/id	<u>±</u> /+	The constant value or the id of an xy series.

Card Type	SPECXVEL			
Description	Specified Velocity,	Specified Velocity, X velocity component		
Required	YES if a feature po	oint or no	de is of Specified Velocity type.	
Format	SPECXVEL cons	stflag	perp-to-boundary value/id	
Sample	SPECXVEL 1	0 138	. 6	
Field	Variable	Value	Description	
1	constflag	0,1	The type code:	
			0 = Constant attribute represented by a single value.	
			1 = Transient attribute represented by xy series.	
2	perp-to-	0,1	1 = Nodal Velocity will be perpenndicular to	
	boundary		boundary.	
3	value/id	±/+	The constant value or the id of an xy series.	

Card Type	SPECYVEL		
Description	Specified Velocity	Y velocit	y component
Required	YES if a feature po	oint or no	de is of Specified Velocity type.
Format	SPECYVEL cons	stflag v	value/id
Sample	SPECYVEL 1	152.6	
Field	Variable	Value	Description
1	constflag	0,1	The type code:
			0 = Constant attribute represented by a single value.
			1 = Transient attribute represented by xy series.
2	perp-to-	0,1	1 = Nodal Velocity will be perpemndicular to
	boundary		boundary.
3	value/id	±/+	The constant value or the id of an xy series.

Card Type	REFINE			
Description	es the size of the	refine poir	nt.	
Required	NO			
Format	REFINE value	REFINE value		
Sample	REFINE 100	REFINE 100		
Field	Variable	Value	Description	
1	Value	+	The constant value which specifies the size of surrounding ele, ments after meshing.	

#### Arcs:

Card Type	SPECHELEV		
Description	Specified Head ele	evation at	tribute
Required	YES if a refine arc	is of Spe	cified Head type.
Format	SPECHELEV con	nstflag	value/Id type total-flow
Sample	SPECHELEV 0	132	0 1
Field	Variable	Value	Description
1	constflag	0,1	The type code:
			0 = Transient attribute represented by xy series.
			1 = Constant attribute represented by a single value.
2	value/id	<u>±</u> /+	The constant value or the id of an xy series.
3	type	0,1	The type code:
			0 = Essential Specified Head Arc.
			1 = Natural Specified Head Arc.
4	total-flow	0,1	The type code:
			0 = Arc is not total flow arc.
			1 = Arc is total flow arc.

Card Type	SPECCONC	SPECCONC		
Description	Specified Concent	tration, Co	oncentration Attribute.	
Required	YES if a feature ar	rc is of Sp	ecified Concentration type.	
Format	SPECCONC const	tflag <sup>.</sup>	value/id	
Sample	SPECCONC 0	1.32		
Field	Variable	Value	Description	
1	constflag	0,1	The type code:	
			0 = Transient attribute represented by xy series.	
			1 = Constant attribute represented by a single value.	
2	value/id	±/+	The constant value or the id of an xy series.	

Card Type	SPECFLOW			
Description	Specified Flow, flo	Specified Flow, flow attribute.		
Required	YES if a feature ar	rc is of Sp	pecified Flow type.	
Format	SPECFLOW cons	tflag	value/id total-flow	
Sample	SPECFLOW 0	1.32	1	
Field	Variable	Value	Description	
1	constflag	0,1	The type code:	
			0 = Transient attribute represented by xy series.	
			1 = Constant attribute represented by a single value.	
2	value/id	<u>±</u> /+	The constant value or the id of an xy series.	
3	total-flow	0,1	The type code:	
			0 = Arc is not total flow arc.	
			1 = Arc is total flow arc.	

Card Type	FLUX			
Description	Flux arc attribute.	Flux arc attribute.		
Required	YES if a feature ar	YES if a feature arc is of Flux type.		
Format	FLUX total-flow			
Sample	FLUX 1			
Field	Variable	Value	Description	
1	total-flow	0,1	The type code:	
			0 = Arc is not total flow arc.	
			1 = Arc is total flow arc.	

### **Polygons:**

Card Type	ADAPTESS			
Description	Adaptive Tesselat	Adaptive Tesselation type attributes.		
Required	YES if a feature po	YES if a feature polygon is of adaptive tessellation type.		
Format	ADAPTESS bias			
Sample	ADAPTESS 1.	ADAPTESS 1.0		
Field	Variable	Value	Description	
1	bias	+	Bias for meshing.	

Card Type	PATCH			
Description	Patch type attribu	Patch type attributes.		
Required	YES if a feature po	YES if a feature polygon is of patch type.		
Format	PATCH bias			
Sample	PATCH 1.0			
Field	Variable	Value	Description	
1	bias	+	Bias for meshing.	

Card Type	CEILING		
Description	Four points definir	ng a ceilin	g.
Required	NO		
Format	CEILING flag	a b	c d
Sample	CEILING 1 0.577 0.577 0.577		
Field	Variable	Value	Description
1	flag	0,1	The type code:
			0 = Ceiling is off.
			1 = Ceiling is on.
2-5	a,b,c,d	±	Four values a,b,c,d for use in plane equation ax
			+ by + cz +d = 0 to define plane of ceiling

Card Type	MAT
Description	Beginning of the polygon material values
Required	NO

Card Type	MN			
Description	Material name and	d id.		
Required	YES if the MAT ca	ard is used	d.	
Format	MN id name	MN id name		
Sample	MN 1 material_1			
Field	Variable	Value	Description	
1	id	+	id of the material.	
2	name	+	name of the material.	

Card Type	MC			
Description	Material colors.			
Required	YES I fthe MAT ca	ard is used	1.	
Format	MN id r g	MN id r g b		
Sample	MN 1 255 255 255			
Field	Variable	Value	Description	
1	id	+	id of the material.	
2-4	r,g,b	+	Red, Green, Blue color components.	

Card Type	MS		
Description	Material pattern.		
Required	YES if the MAT ca	rd is used	d.
Format	MN id patte:	rn	
Sample	MN 1 5		
Field	Variable	Value	Description
1	id	+	id of the material.
2	pattern	+	Pattern index.

## 14.11.4 Drawing Objects

The following cards are used to define rectangles, ovals, lines, and text strings.

Card Type	RECT
Description	The beginning of a set of cards defining a rectangle.
Required	NO

Card Type	OVAL
Description	The beginning of a set of cards defining an oval.
Required	NO

Card Type	C# (C1, C2, C3, C4, for the four corner points)			
Description	Corner point identifiers of rectangle or oval.			
Required	YES if a rectangle	YES if a rectangle or oval has been defined.		
Format	C1 x y z			
Sample	C1 10 10 0			
Field	Variable Value Description			
1-3	х, у, z	±	Coordinates of corner point.	

Card Type	THICK			
Description	Line thickness ide	Line thickness identifier.		
Required	YES if a line, recta	YES if a line, rectangle or oval has been defined.		
Format	THICK width			
Sample	THICK 1			
Field	Variable Value Description		Description	
1	width	+	Line thickness in pixels.	

Card Type	STYLE				
Description	Line style identifie	Line style identifier.			
Required	YES if a line, recta	angle or o	val has been defined.		
Format	STYLE style	STYLE style			
Sample	STYLE 0	STYLE 0			
Field	Variable	Value	Description		
1	style	+	0 - Solid line style.		
			1 - Dashed line style.		

Card Type	LINECOL			
Description	Line color identifier.			
Required	YES if a line, recta	YES if a line, rectangle or oval has been defined.		
Format	LINECOL r g b			
Sample	LINECOL 255 2	LINECOL 255 255 255		
Field	Variable	Value	Description	
1-3	r, g, b	0-255	Red, green and blue color components	

Card Type	FILLCOL			
Description	Polygon fill color id	Polygon fill color identifier.		
Required	YES if a rectangle	or oval ha	as been defined.	
Format	FILLCOL r g b			
Sample	FILLCOL 255 2	FILLCOL 255 255 255		
Field	Variable	Value	Description	
1-3	r, g, b	0-255	Red, green, blue color components	

Card Type	FILLPAT			
Description	Polygon fill pattern	Polygon fill pattern identifier.		
Required	YES if a rectangle	YES if a rectangle or oval has been defined.		
Format	FILLPAT pattern			
Sample	FILLPAT 0	FILLPAT 0		
Field	Variable	Value	Description	
1	pattern	+	Pattern index.	

Card Type	LINE
Description	Beginning of a set of cards defining a line object.
Required	NO

Card Type	VERTS			
Description	Number of points	Number of points in a line.		
Required	YES if a line has b	een defin	ed.	
Format	VERTS count			
Sample	VERTS 3	VERTS 3		
Field	Variable	Value	Description	
1	count	+	Number of points in the line.	

Card Type	PT				
Description	Defines a point on	Defines a point on a line object			
Required	YES if a line has b	een defin	ed.		
Format	PT x y z	РТхуг			
Sample	PT 213.2. 523	PT 213.2. 523.2 0			
Field	Variable	Value	Description		
1-3	хуг	±	Coordinates of the point.		

Card Type	ARRHED			
Description	Arrow head type id	dentifier.		
Required	YES if a line has b	een defin	ed.	
Format	ARRHED style			
Sample	ARRHED 0			
Field	Variable	Value	Description	
1	style	0	No arrow head.	
		1	Arrow head at beginning of line. Arrow head at end of line.	
		2	Arrow heads at both ends of line.	
		3		

Card Type	HEDWID			
Description	Arrow head base	Arrow head base width identifier.		
Required	YES if a line has b	een defin	ed.	
Format	HEDWID width	HEDWID width		
Sample	HEDWID 10	HEDWID 10		
Field	Variable	Value	Description	
1	width	+	Width of the base of the arrow head in pixels.	

Card Type	HEDLEN			
Description	Arrow head length	Arrow head length identifier.		
Required	YES if a line has b	een defin	ed.	
Format	HEDLEN length			
Sample	HEDLEN 25	HEDLEN 25		
Field	Variable	Value	Description	
1	length	+	Length of the arrow head in pixels.	

Card Type	TEXT
Description	Beginning of a set of cards defining a text object.
Required	NO

Card Type	STRING				
Description	Text string identifie	Text string identifier.			
Required	YES if a text string	YES if a text string has been defined.			
Format	STRING "string	STRING "string"			
Sample	STRING "map to	STRING "map title"			
Field	Variable	Value	Description		
1	string	str	Text string.		

Card Type	LOCAL				
Description	Text string location	Text string location identifier.			
Required	YES if a text string	has beer	n defined.		
Format	LOCAL x y				
Sample	LOCAL 100 200	LOCAL 100 200			
Field	Variable	Value	Description		
1-2	х, у	±	Coordinates of the beginning of the text string.		

Card Type	PCFONT				
Description	PC font identifier.	PC font identifier.			
Required	YES if a text string	YES if a text string has been defined.			
Format	PCFONT id				
Sample	PCFONT 2				
Field	Variable	Value	Description		
1	id	+	The id of the PC font.		

Card Type	UNIXFONT				
Description	UNIX font identifie	UNIX font identifier.			
Required	YES if a text string	has beei	n defined.		
Format	UNIXFONT id				
Sample	UNIXFONT 2	UNIXFONT 2			
Field	Variable Value		Description		
1	id	+	The id of the UNIX font.		

## 14.12 Image Files

Image files are used in conjunction with TIFF files which have been previously imported to SMS and registered. They include the name of the TIFF file, the registration points, and the bounds of the clipping window. The format of the image file is shown in Figure 18.50 and a sample image file is shown in Figure 18.51.

```
IMAGE /* File type identifier */
TIFF "filename" /* Indicates the name of the tiff file used */
IMREGPTS
PT1 ul v1 x1 y1
PT2 u2 v2 x2 y2
PT3 u2 v2 x2 y2
CLIPPOINT
x1 x2
y1 y2
```

Figure 14.16 The Image File Format.

```
IMAGE
TIFF "easttex.tif"
IMREGPTS
PT1 117 797 0.000000 10000.000000
PT2 117 88 0.000000 0.000000
PT3 1053 88 13220.000000 0.000000
CLIPPOINT
-1082.059503 13885.402536
-992.568818 8457.158566
```

Figure 14.17 Sample Image File.

The card types used in the Image file format are as follows:

Card Type	IMAGE
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	TIFF	TIFF			
Description	Defines the name	Defines the name of the TIFF file to be displayed as an image.			
Required	YES	YES			
Format	TIFF "filename	TIFF "filename"			
Sample	TIFF "easttex	TIFF "easttex.tif"			
Field	Variable	Value	Description		
1	filename	str	The name of the TIFF file.		

PT1, PT2, PT3				
The three registration points used to define locations on a given image.				
YES	YES			
PT1 tx1 ty1 w	x1 wy1			
PT2 tx2 ty2 w	x2 wy2			
PT3 tx3 ty3 wx3 wy3				
PT1 117 797 0.000000 10000.000000				
Variable Value		Description		
tx ty	±	Texture map coordinates.		
wx wy	±	World coordinates.		
	The three registrat YES PT1 tx1 ty1 w PT2 tx2 ty2 w PT3 tx3 ty3 w PT1 117 797 0 PT2 117 88 0.0 PT3 1053 88 13 Variable tx ty	The three registration points YES  PT1 tx1 ty1 wx1 wy1 PT2 tx2 ty2 wx2 wy2 PT3 tx3 ty3 wx3 wy3 PT1 117 797 0.000000 PT2 117 88 0.000000 PT3 1053 88 13220.0  Variable Value tx ty wx wy		

Card Type	CLIPPOINTS			
Description	Defines the coordinates of the area in the TIFF file to be displayed as the image.			
	(The area clipped	and displa	ayed from the TIFF file.	
Required	YES			
Format	CLIPPOINTS			
	xmin xmax			
	ymin ymax			
Sample	CLIPPOINTS			
· ·	-628.990382 14			
	-857.665608 83	354.617	436	
Field	Variable	Value	Description	
1-2	xmin xmax	±	Min and max values in the x direction.	
3-4	ymin ymax		Min and max values in the y direction.	
		±	This and mak raides in the y and show	

## 14.13 Mesh From Polygon Files

Mesh from polygon files are used with the *Mesh From Poly* command. It is accessed from the *Elements* menu while in the *Mesh Module*. The format for an adaptive tessellation file is identical to that of a polygon file, except that additional feature information describing channel or ridge locations may be included. These include *refine points* and *breaklines*.

An adaptive tessellation file contains one or more POLY cards as shown in Figure 14.18. The vertices of the polygon are listed in counter-clockwise order. In some cases, it is necessary to define a polygon with holes in the interior using a polygon file. In such cases, the boundary of the polygon should be defined with a POLY card with the vertices in counter-clockwise order. Each of the interior holes should then be listed with separate POLY cards with the vertices listed in clockwise order.

A refine point causes *SMS* to insert a hard nodal point into the mesh. When the adaptive tessellation is run, *SMS* creates six small triangles surrounding the refine point. Each of the triangles will have edge lengths corresponding to a specified value. A smooth element size transition will be defined as the mesh moves out from the refine point. The smaller the edge length, the smaller the elements surrounding the refine point. A breaklinepolygon causes *SMS* to force a breakline through the mesh. Element edges will conform to the breakline.

Figure 14.18 Adaptive Tessellation File Format.

The format of the adaptive tessellation file is as follows:

Card Type	POLY	POLY			
Description	Defines one co	Defines one complete loop of a polygon.			
Required	YES				
Format	POLY np x1 y1 z1 x2 y2 z2 xnp ynp znp				
Sample	POLY 4 0.0 0.0 0.0 10.0 0.0 0.0 10.0 10.0 0.0 0.0 10.0 0.0				
Field	Variable	Value	Description		
1	np	+	The number of points in the polygon loop.		
2-4	x,y,z	±	Vertex coordinates. Repeat for each vertex. List in counter-clockwise order for the outer polygon boundary. List in clockwise order for holes in the interior of a polygon.		

Card Type	REFPT			
Description	Defines a refine point within a 2D mesh			
Required	NO			
Format	REFPT x y z edgelen			
Sample	REFPT 10.0 25.0 13.5 0.25			
Field	Variable	Value	Description	
1-3	x,y,z	±	Location of the refine point within the boundary polygon.	
4	edgelen	±	length of one of the edges of the triangles that will immediately surround the refine point	

Card Type	BLINE		
Description	Defines a feat	ure line	e or "breakline" within a 2D mesh
Required	NO		
Format	BLINE np x1 y1 z1 x2 y2 z2 xnp Ynp znp		
Sample	BLINE 4 1.0 0.0 0.0 5.0 1.0 2.0 10.0 5.0 2.5 12.0 10.0 3.0		
Field	Variable	Value	Description
1	np	+	The number of vertices in the break line.
2-4	х,у,z	±	Vertex coordinates. Repeat for each vertex. The breakline should not cross itself.

#### 14.14 XY Series Files

The XY Series Editor described in Chapter 13 is used in several places in SMS, including the definition of time dependent boundary conditions and rating curves. The XY Series Editor is a general purpose editor for entering curves or pairs of lists of data. The XY Series Editor allows a curve to be imported from a file, created and edited graphically, or created and edited using two columns of edit fields in a spreadsheet-like interface.

XY series files can be used to input a set of curves into the XY Series Editor. XY series files are also used to export curves generated within the Editor for future use.

The format of the XY series file is shown in Figure 14.19. Curves are defined in an XY Series File using one of three types of cards: XY1, XY2, or XY3. With the XY1 card, both the x and y values are listed for each point on the curve. There is no limit to the spacing or interval used between subsequent x values. The XY2 card is identical to the XY1 card except that the number of points and the x values are assumed to be static and cannot be altered by the user. With the XY3 card, the x values are defined by a beginning x value, an initial increment in x, and a per cent change in x per increment. Only the y values are explicitly listed.

```
XY1 id n dx dy rep begc name
                                                /* XY Series vers. #1 */
x1 y1 /* XY values */
x2 y2
xn yn
XY2 id n dx dy rep begc name
                                                /* XY Series vers. #2 */
       /* XY values */
х1 У1
×2 У2
xn yn
XY3 id n x1 incx pcx dx dy rep begc name
                                                /* XY Series vers. #2 */
У1
       /* Y values */
У2
```

Figure 14.19 The XY Series File Format.

The card types used in the XY series file format are as follows:

Card Type	XY1				
Description	Defines a curve with a list of XY values. Any number of points and any x spacing between points may be used.				
Required	NO				
Format	XY1 id n dx dy rep begc name				
	x1 Y1				
	x2 y2				
	•				
	x <sub>n</sub> y <sub>n</sub>				
Sample		0 head			
Sample	0.0	0.0			
	1.0	2.0			
	2.5 3.0	7.0 8.0			
	4.5	9.5			
Field	Variable	Value	Description		
1	id	+	The ID of the XY series.		
2	n	+	The number of point in the series.		
3	dx	0,1	A flag defining whether the x values listed are to be		
			interpreted as incremental (dx=1) or absolute (dx=0).		
4	dy	0,1	A flag defining whether the y values listed are to be		
			interpreted as incremental (dy=1) or absolute (dy=0).		
5	rep	0,1	A flag defining whether the xy series is to be		
			interpreted as cyclic (repeating).		
6	begc	±	The x value in the series where the cyclic portion of		
			the curve begins. Value is ignored if rep=0.		
7	name	str	The name of the series.		
8-9	x,y	±	The xy values of the points defining the curve. Repeat		
			n times.		

Card Type	XY2
Description	Defines a curve with a list of XY values. This card is identical to the XY1 card except
	that the number of points and the x values are assumed to be static and cannot be altered by the user.
	altered by the user.

Card Type	XY3		
Description	Defines a curve with a list of Y values. The x values are		
		eginnin	g value, an increment, and a bias.
Required	NO		, ,
Format		cx blas	x dx dy rep begc name
	У1		
	У2		
	Уn		
Sample	XY3 1 10 0 1 0	0 0 0	0 head
Campio	0.0		
	2.0		
	8.0		
	9.5		
	9.1		
Field	Variable	Value	Description
1	id	+	The ID of the XY series.
2	n	+	The number of point in the series.
3	x1	±	The first x value.
4	incx	±	The increment in x used to compute the next x value.
5	pcx	+	The per cent change in x used to compute subsequent
			x values. Expressed as a decimal, i.e., 0.05 = 5%.
6	dx	0,1	A flag defining whether the x values listed are to be
			interpreted as incremental (dx=1) or absolute (dx=0).
7	dy	0,1	A flag defining whether the y values listed are to be
			interpreted as incremental (dy=1) or absolute (dy=0).
8	rep	0,1	A flag defining whether the xy series is to be
			interpreted as cyclic (repeating).
9	begc	±	The x value in the series where the cyclic portion of
			the curve begins. Value is ignored if rep=0.
10	name	str	The name of the series.
11	У	±	The y values of the points defining the curve. Repeat n
			times.

### **14.15 TIN Files**

TIN files store triangulated irregular network data. Multiple TINs can be stored to a single file. The TIN file format is shown in Figure 14.20. *SMS* can import TINs, converting them to triangular elements.

```
TIN
           /* File type identifier */
          /* Beginning of TIN group */
TNAM name
                                           /\bar{*} Name of TIN */
MAT id /* TIN material id */
VERT nv /* Beg. of vertices */
x1 y1 z1 lf1
                                           /* Vertex coords. */
x2 y2 z2 lf2
x<sub>nv</sub> y<sub>nv</sub> z<sub>nv</sub> lf<sub>nv</sub>
TRI nt /* Beg. of triangles */
v11 v12 v13
                                         /* Triangle vertices */
v21 v22 v23
v<sub>nt1</sub> v<sub>nt2</sub> v<sub>nt3</sub>
ENDT /* End of TIN group */
/* Repeat TIN group for other TINs */
```

Figure 14.20 TIN File Format.

The cards used in the TIN file are as follows:

Card Type	TIN
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	BEGT
Description	Marks the beginning of a group of cards describing a TIN. There should be a corresponding ENDT card at a latter point in the file. No fields.
Required	YES

Card Type	TNAM		
Description	Provides a name to be associated with the TIN.		
Required	NO		
Format	TNAM name		
Sample	TNAM bathymetry		
Field	Variable Value Description		
1	name	str	The name of the TIN.

Card Type	MAT			
Description			id with the TIN. This is typically the id is below the TIN.	
Required	NO			
Format	MAT id	MAT id		
Sample	MAT 3			
Field	Variable	Value	Description	
1	id	+	The material ID.	

Card Type	VERT			
Description	Lists the vert	Lists the vertices in the TIN		
Required	YES			
Format	VERT nv x1 y1 z1 lf1 x2 y2 z2 lf2 xnv ynv znv lfnv			
Sample	VERT 4 0.0 3.1 7.8 0 5.3 8.7 4.0 1 2.4 4.4 9.0 1 3.9 1.2 3.6 0			
Field	Variable	Value	Description	
1	nv	+	The number of vertices in the TIN.	
2-4	x,y,z	±	Coordsinates of vertex.	
5	lf	0,1	Locked / unlocked flag for vertex (optional). 0=unlocked, 1=locked. Repeat fields 2-5 nv times.	

Card Type	TRI		
Description	Lists the tria	ngles ir	n the TIN
Required	NO		
Format	TRI nt V11 V12 V13 V21 V23 V23 Vnt1 Vnt2 Vnt3		
Sample	TRI 4 5 1 4 4 1 2 4 2 3 5 4 3		
Field	Variable	Value	Description
1	nt	+	The number of triangles in the TIN.
2-4	v1,v2,v3	+	Vertices of triangle listed in a counter-clockwise order. Repeat nt times.

Card Type	ENDT
Description	Marks the end of a group of cards describing a TIN. There should be a corresponding BEGT card at a previous point in the file. No fields.
Required	YES

### 14.16 Material Files

Each element of a 2D mesh has an assigned material ID. Specific material properties are related to the analysis models, and are stored in the analysis files. However, general material properties, such as color, are not stored in these files. Therefore, they are stored in the material file. A material ID represents an index to a global list of materials. The material file associates general attributes such as a name, color, and pattern with each of the materials. The format for a material file is shown in Figure 14.21.

```
MAT /* File type identifier */
MN id name /* Material name */
MC id red green blue /* Material color */
MS id stippleid /* Material stipple (fill pattern) */
```

Figure 14.21 Material File Format.

Each card in the material file represents an attribute for a material. The attribute cards can be repeated as many times as necessary to define each material being used. The cards used in the material file are as follows:

Card Type	MAT
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	MN		
Description	Identifies a name to be associated with the material.		
Required	NO		
Format	MN id name		
Sample	MN 5 bedrock		
Field	Variable	Value	Description
1	id	+	The ID of the material.
2	name	str	The name of the material.

Card Type	MC			
Description	Identifies a color to be associated with the material.			
Required	NO			
Format	MN id red gree	MN id red green blue		
Sample	MN 5 124 67 245			
Field	Variable	Value	Description	
1	id	+	The ID of the material.	
2	red	0-255	The value of the red component of the color.	
3	green	0-255	The value of the green component of the color.	
4	blue	0-255	The value of the blue component of the color.	

Card Type	MS		
Description	Identifies a stipple (fill pattern) to be associated with the material. This stipple is used whenever an object is being drawn using color filled polygons.		
Required	NO		
Format	MN id stippleid		
Sample	MN 5 13		
Field	Variable Value Description		
1	id	+	The ID of the material.
2	stippleid	+	The ID of the stipple.

## 14.17 XYZ Files

XYZ points can be imported from an ASCII file and converted to nodes. This provides a convenient way to import a set of points for mesh construction operations. The format of the XYZ file is shown in Figure 14.22.

```
XYZ (The first line should contain the word "XYZ")
x1 y1 z1 (Listing of XYZ data point coordinates)
x2 y2 z2
x3 y3 z3
.
.
.
.
xn yn zn
```

Figure 14.22 XYZ File Format.

# References

Clough, R. W., and J. L. Tocher, 1965, Finite element stiffness matrices for analysis of plates in bending, *Proc. Conf. Matrix Methods in Structural Mechanics, Wright-Patterson A.F.B.*, Ohio, Air Force Flight Dynamics Lab., Research and Technology Division, Air Force Systems Command, The Air Force Institute of Technology, Air University, pp. 515-545.

Davis, J.C., 1986, *Statistics and Data Analysis in Geology*, John Wiley & Sons, New York, 550 p.

Deutsch, C.V., & A.G. Journel, 1992, *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York, 340 p.

Franke, R. & G. Nielson, 1980, "Smooth interpolation of large sets of scattered data," *International Journal for Numerical Methods in Engineering*, Vol. 15, pp. 1691-1704.

Franke, R., 1982, "Scattered data interpolation: tests of some methods," *Mathematics of Computation*, Vol. 38, No. 157, pp. 181-200.

Heine, G. W., 1986, "A controlled study of some two-dimensional interpolation methods," *COGS Computer Contributions*, Vol. 2, No. 2, pp. 60-72.

Jones, N. L., 1990, Solid Modeling of Earth Masses for Applications in Geotechnical Engineering, Ph.D. Dissertation, The University of Texas at Austin, 324 p.

Journel, A.G., & Huijbregts, C.J., 1978, *Mining geostatistics*. Academic Press, New York, NY.

Lam, N.S., 1983, "Spatial interpolation methods: a review," *The American Cartographer*, Vol. 10, No. 2, pp. 129-149.

Lancaster, Peter and Kestutis Salkauskas, 1986, *Curve and Surface Fitting*, Academic Press, London, 280 pp.

McDonald, M.G., & A.W. Harbaugh, 1988, A modular three-dimensional finite-difference ground-water flow model, Techniques of Water Resources Investigations 06-A1, United States Geological Survey.

Olea, R.A., 1974, "Optimal contour mapping using universal kriging." *J. Geophys. Res.*, Vol. 79, No. 5, pp. 695-702.

Owen, S.J., 1992, An implementation of natural neighbor interpolation in three dimensions, Master's Thesis, Brigham Young University, 119 p.

Philip, G.M., & D.F. Watson, 1986, "Comment on 'comparing splines and kriging," *Computers and Geosciences*, Vol. 12, No. 2, pp. 243-245.

Royle, A. G., F. L. Clausen, & P. Frederiksen, 1981, "Practical universal kriging and automatic contouring," *Geo-Processing*, Vol. 1, No. 4, pp. 377-394.

Shepard, D., 1968, "A two dimensional interpolation function for irregularly spaced data," *Proc. 23rd National Conference of the ACM*, pp. 517-523.

Sibson, R., 1981, "A brief description of natural neighbor interpolation," *Interpreting Multivariate Data*, John Wiley & Sons, New York, pp. 21-36.

Watson, D. F. and G. M. Philip, 1985, A refinement of inverse distance weighted interpolation, *Geo-Processing*, Vol., 2, No. 4, pp. 315-327.

WES, 1994, FEMWATER Reference Manual, U.S. Army Engineer Waterways Experiment Station.

Yeh, G.T., S.S. Hansen, B. Lester, R. Strobl, J. Scarbrough, 1992, 3DFEMWATER/3DLEWASTE: Numerical Codes for Delineating Wellhead Protection Areas in Agricultural Regions Based on the Assimilative Capacity Criterion, U.S. Environmental Protection Agency.

Zheng, C., 1990, "MT3D: A ModularThree-Dimensional Transport Model for Simulation of Advection, Dispersion and Chemical Reactions of Contaminants in Groundwater Systems." S.S. Papadopulus & Associates, Inc.

# Index

1D12-1	barycentric	5-12
2D Boundary Fitted Grid	bias	4-11
module1-4	boundary condition	
2D Mesh	delete	8-6
from feature objects6-20	HIVEL	10-4
import2-10	node	8-2
module1-4	nodestring	8-4
2D Scatter Point	open	8-2
file2-7, 3-3, 14-5	save	8-2
module1-4	breakline	4-22, 14-43
XY format2-7	add	4-22
XYD format2-7	CAD	6-1
active	card	14-1
data set5-5	\$L	8-11
gage3-16	\$M	8-12
scatter point set5-2, 5-3	BCC	8-12
xy series13-2	DE	8-13
adaptive tessellation	FT	8-11
animation3-4, 3-11, 3-13	GS	8-12
flow trace3-13	LA	8-12
time3-13	PE	8-13
append2-9	SI 8-11	
arc6-3	T1-3	8-10
create6-5	TI 8-11	
dangling6-7	TR	8-13
intersecting6-6	TZ	8-11
select6-4	circumcircle	4-15
ASCII output8-13	clean	
attributes	feature objects	6-6
element4-27	Clough-Tocher	
mesh4-27	color	•
background2-20	background	2-20

contour3-8	animation	3-13
HSV 3-9	calculator	3-4, 3-5
intensity 3-9	contours	
ramp3-9	deleting	
scatter point 5-3	difference between two	4-6
concentration3-15	elevation	3-4
conceptual model	export	3-3
map to 2D mesh6-22	file3-2, 14-17, 14	
continutity string8-6	file (ASCII)	14-8
contour	file (binary)	
animation3-13	import	
colors 3-9	info	
label 4-5	legend	
label tool	maximum	
labels3-10	mean	
legend 3-10	minimum	
spline	name	
values	scalar file (binary)	
convert	scalar file(ASCII)	
mesh4-5	standard deviation	
scatter point5-4	statistics	
coordinates	time step	
barycentric	vector	
gage	vector file (ASCII)	
local	vector file (binary)	
nodes	DC records	
snapping to a grid	delete	
xy series	boundary condition	
copyrighti	confirm	
coverage	data set	
2D mesh attributes 6-21	DXF objects	
2D mesh type	element	
• •	gage	
copy 6-10 elevation 6-10	images	
inactive color	node	
	demo	
new		2-13
options	display	2.20
types	background	2-20
visibility6-11	contour	
create	frame image	
element	grid	
gage4-3	refresh	
node4-1	vectors	
nodestring	window bounds	2-19
DA records	display options	
data1-5, 11-11	DXF files	
browser 3-2	feature objects	
film loops	FESWMS	
menu 3-1	gage curve	
mesh4-5	gages	
vectors3-6	HIVEL	
data set	mesh	,
active3-4, 5-5	nodes	4-2

RMA28-13	material type	4-5
scatter points5-3	merge	
SED2D-WES9-9	merge/split	
DM card8-13	operations	
DP records12-8	options	
dragging	quadratic	
gages4-3	quadrilateral	
nodes4-2	refine	
drawing objects	relax	
create ellipse6-24	renumber	
create line6-24	select	
create rectangle6-24	smooth	
create text6-24	split	
default attributes6-26	swap edges	
line attributes6-26	triangular	
move to back6-27	types of	
move to front6-27	elevation	
reading6-35	enabling SMS	
rectangle/ellipse attributes6-25	environment	2 10
saving	save	2-11
selecting6-24	exit	
shuffle down	export	2-1.
shuffle up6-27	data set	3_3
text attributes	file	
tools	gage	
DXF	scatter points	
Converting to feature objects6-34	WKS	
converting to scatter points6-35	xy series	
deleting objects6-35	extrapolation	
display options6-34	default value	
import	inverse distance weighted	
importing	natural neighbor	3-18
DXF -> Feature Objects	feature	4
DXF -> Scatter Point6-35	from material	4-6
ECGL	feature object attribute	c 15
<i>fax</i> i	continuity check	
phonei	refine points	
edit	specified concentration	
node	specified flow	
WSPRO section	specified head	
EDIT MENU2-16	specified velocity	
element	total flow	6-17
breakline4-22	feature objects	
concave4-5	arcs	
conversion4-24	attributes	
convert4-24	build polygon	
create4-4	clean	
creating4-4	convert node to vertex	
deletion4-5	convert vertex to node	
find4-21	Converting from DXF	
ill-formed	coverages	
linear4-4	create arc	
material4-27	create point	6-5

create vertex6-5	image2-7, 14-41
creating 2D meshes6-20	import2-9
dangling arcs6-7	map14-29
definition	material2-7, 2-17, 14-50
display options6-11	menu2-7
intersecting arcs	mesh from polygon14-43
map to 2D mesh6-22	RMA23-3
nodes6-3	scalar data set (ASCII) 14-17, 14-21
points 6-2	scatter point5-2
polygons 6-3	SED2D-WES9-1
reading 6-35	spreadsheet3-18
redistribute vertices 6-7	types2-7
saving6-35	vector data set (ASCII)14-19
select arc 6-4	vector data set (binary)14-23
select point/node6-4	xy series14-45
select polygon	XYZ14-52
select vertices	film loop
snap nodes	data set
types	dialog
vertices 6-3	file
feature objects attribute 6-13	flow trace
FESWMS	image size
append	playback
Display Options	saving3-14
FESWMS Boundary Conditions	setup
FESWMS Boundary Section	time
FESWMS Ceiling	film loops
FESWMS Control 11-13	find
FESWMS Culvert 11-10	duplicates4-13
	element
FESWMS Drop Inlet	
FESWMS Flux String	node
FESWMS Initial Conditions	FL Card
FESWMS Material Properties	FL records
FESWMS Model Check	format
FESWMS Pier	reference manual1-6
FESWMS Weir	frame image2-19
FESWMS Wind Conditions	fringe
FHWA 12-1	colors3-9
field data	legend
file	gage3-15
2D scatter point2-7, 3-3, 14-5	create4-3
append2-9	creating
binary	curves
cards	deleting3-17
data set	editing3-18
data set (ASCII)	file 3-16, 14-25
data set (binary)	import3-16
DXF2-11	interpolation method3-16
export2-10	plot manager3-18
FESWMS	plotting3-20
film loop 3-14	printing3-18
gage3-16, 14-25	select4-3
HIVEL2D 3-3	selecting3-17

tools	3-17	registering	6-28
GC String	8-6	registering tools	
geometry		resampling	
append	2-10	saving	
open		tiff	
save		import	,
get info	2-11	xy series	13-2
GFGEN		import	
append	2-10	file	
GIS		index	
from material		element	4-21
GR Card		material	
GR records		node	
gradient		scatter point	
graph		interpolation	
grid		cluster	
snap to		global	
help		gradient	
HIVEL		IDW(see inver	
boundary condition			(see inverse distance weighted)
delete	10-5	linear	=
boundary condition		local	
card		local coordinates	
grav	10-7	natural neighbor	
iter		nodal functions	
mcon		quadratic	
mtyp		Shepard's method(see inve	
pgwc		subsets	_
SI 10-7		truncation	
step	10-7	inverse distance weighted	
T1-T3		barycentric weights	
time		interpolation subsets	
turb		local weighting method	
display options		local/global	
hot start		nodal functions	
materials		weights	
model check		iterations	
new		legend	
open		lock/unlock	
run control		node	4-13
save	10-2	mage	
title	10-6	file	2-7
units control	10-7	map	
HP records		feature objects	6-2
IDW(see inverse dis	tance weighted)	file	
image file	_	module	
images		material	
delete		dialog	2-17
export tiff	6-32	file	
exporting tiff		mesh	
fit entire image to screen		materials	
import		HIVEL	10-8
reading	6-35	RMA2	8-7

maximum	. 3-7	natural neighbor	
mean	. 3-4	bounding window	
menu		extrapolation	5-18
bar	. 2-1	local coordinates	5-16
general	. 2-6	weights	5-17
build mesh	4-15	ND records	
data	. 3-1	new	
display2-18,	2-19	HIVEL	
edit		SED2D-WES	
file		WSPRO	
interpolation		nodal function	
node		node	
merge		boundary condition	1 12
merge/split4-23,		boundary condition	
mesh		coincident	
convert		convert to vertex	
create		create	
creating		delete	
data		distance between	
display options2-19	, 4-7	duplicates	4-13
elements	4-16	edit	I-13, 6-4
generation4-5	, 4-8	editing	4-2
interpolating to	. 5-4	feature objects	6-3
material	4-27	find	4-12
transform	4-14	insertion	
mesh from polygon		interpolation	
file	4-43	lock/unlock	
mesh generation	5	midside	
adaptive tessellation6-19,	6-22	operations	
patch		options	
paving		renumber	
rectangular patches		select	
triangular patches		snapping together	6-6
triangulation4-4,		nodestring	0.4
minimum	. 3-7	boundary condition	
model check		continutity string	
HIVEL		create	
RMA2		select	
SED2D-WES		open	
WSPRO 1	2-14	boundary condition	8-2
module	. 1-4	geometry	8-2
2D Boundary Fitted Grid	. 1-4	HIVEL	10-2
2D Mesh	. 1-4	SED2D-WES	9-2
2D Scatter Point	. 1-4	WSPRO	12-2
map		orientation	
mesh		paper	2-12
river		OVERVIEW	
scatter point	,	page layout	
selecting a new		page size	
•			∠-1∠
move to back		palette	2.4
move to front		dynamic	
MS Windows		module	
N records	12-3	static	2-3

pan2-3	display options	8-13
patch6-19	drying	
Clough-Tocher5-13	files	
Coon's patch4-18	geometry	
rectangular4-18	global bc	
triangular4-19	iterations	
paving6-23	machine type	
pixel map3-11	materials	
plot	model check	
window2-5	open bc	
plot options3-20	open geometry	
point6-2	other options control	
create6-5	peclet control	
edit6-4	save bc	
select6-4	save geometry	
polygon2-16	time control	
build6-6	title	
select6-5	units control	
Thiessen5-15	wetting	
polygons	roughness	
feature objects6-3	run control	12 11
postscript	HIVEL	10-6
setup2-12	RMA2	
Print 2 12	SA Card	
SED2D-WES9-8	SA records	
printer	save	
postscript2-11	boundary condition	
setup2-12	environment	
printing2-11	geometry	
gage plots	HIVEL	
page layout	SED2D-WES	
postscript2-12	WSPRO	
setup2-12	scalar	12-2
quadrilateral	displaying	2.6
split4-24	scatter point	
•	active	
quit		
redistribution 6-7	color	
	convert	,
linear interpolation6-8	create	
spline	data set	
reference	display options	
format	editing	
references	file	
refine	icon	
refine point	interpolation	
refresh	numbers	
register	read	
relax	saving to a file	
renumber	selecting	
river	set	
module1-5	symbols	
window12-2	scour	12-8
RMA2 8-1	section	

edit 12-3	exporting	6-33
SED2D-WES 9-1	import	
BC Concentrations9-7	memory requirements	
Display Options 9-9	registering	
File9-1	registering tools	
Global Parameters 9-2	resampling	
Local Parameters	time	
Model Check	animation	
Model Control9-7	TIN	
new9-2	import	
open 9-2	tool	-
Print Control 9-8	general palette	2-2
save9-2	mesh palette	
select	scatter point palette	
all2-16	total flow	
arc 6-4	transformation	
by material2-16	triangle	
element4-5	merge	4-23
gage3-17, 4-3	triangulated irregular network	
node	triangulation	2 10
node	Delauney	5-15
nodestring	mesh	
point	truncation	1 <i>J</i>
polygon6-5	interpolation	5_5
rectangular patch	units	
scatter point	HIVEL	
scatter point set	Unix	
triangular patch 4-20	vector	1-0
vertices	animation	2 12
	velocity	
with poly	verification	
xy series point		3-13
shading	vertex	6.7
color fill contours	convert to node	
shuffle down	create	
shuffle up	edit	
smooth	redistribution	
snap	select	
to grid	vertices	
spline	feature objects	
split	water surface elevation	
spreadsheet	WES	
SRD	wetting/drying	8-13
standard deviation	window	
super file	bounds	
SMS 14-2	edit	
symbol	film loop	
scatter point 5-3	gage plot	
tension3-8	graphics	
Thiessen polygon 5-15	help	
ΓIFF	plot	2-1, 2-5
import 2-9	river	
tiff images6-1, 6-28	tool palette	2-1
export 6-32	WKS	3-18

WSPRO12-1	translate	12-4
display options12-14	SRD	12-4
FL12-5	xy series	
GR12-4	active	13-1
model check12-14	compress	
new12-2	creating point	
open12-2	delete	
parameters12-12	duplicate	
records	editing	
DA12-8	editor	
DC12-8	export	13-2
DP12-8	file	
FL12-3	frame	
GR12-3	import	
HP12-5	insert	
N 12-3	interpolate	
ND12-3	list	
SA12-3	pan	
roughness12-11	plot	
SA12-5	selectingpoint	
save12-2	update	
scour12-8	xy options	
section	zoom	
attributes12-5	XYZ	
bridges12-6	file	14-52
cross sections12-5	import	
edit12-3	zoom	
guide bank12-9	z-value	
road12-8	interpolate	4-9
sulvert12-9	<b>.</b>	